Empir Software Eng (2017) 22:1103–1142
DOI 10.1007/s10664-016-9479-8

Tackling the term-mismatch problem in automated trace retrieval

Jin Guo[1] · Marek Gibiec[2] · Jane Cleland-Huang[1]

Published online: 23 November 2016
© Springer Science+Business Media New York 2016

Abstract Software systems operating in any type of safety or security critical domains must comply with an increasingly large and complex set of regulatory standards. Compliance is partially demonstrated through establishing trace links between requirements and regulatory codes. Such links can be constructed manually or through semi-automated techniques in which the text in the regulatory code is used to formulate an information retrieval query. However, trace retrieval solutions are not effective when significant vocabulary mismatches exist between regulatory codes and product level requirements. This paper describes and compares three query augmentation techniques for addressing the term mismatch problem and improving the quality of trace links generated between regulatory codes and requirements. The first trains a classifier to replace the original query with terms learned from a training set of regulation-to-requirements trace links. The second, replaces the original query with terms learned through web-mining; and the third utilizes a domain ontology to augment query terms. The ontology is constructed manually using a guided approach that leverages existing traceability knowledge. All three techniques were evaluated against security regulations from the USA government's Health Insurance Privacy and Portability Act (HIPAA) traced against ten healthcare related requirements specifications. The classification approach returned the best results; however, improvements were observed with both the classification and ontology based solutions. The web-mining technique showed improvements in only a subset of queries. The three query augmentation techniques offer tradeoffs in terms of performance, cost and effort, and usage viability within a specific project context.

Communicated by: Patrick Mäder, Rocco Oliveto and Andrian Marcus

✉ Jane Cleland-Huang
JaneClelandHuang@nd.edu

Jin Guo
jguo3@nd.edu

Marek Gibiec
mgibiec@gmail.com

[1] University of Notre Dame, Notre Dame, IN, USA

[2] DePaul University, Chicago, IL, USA

Keywords Requirements engineering · Traceability · Query augmentation · Semantic traceability

1 Introduction

Software intensive systems, especially those that are deployed in safety or security critical applications, must conform to an increasingly large and complex set of regulatory codes. For example all health-care related products in the USA are governed by the Health Insurance Portability and Accountability Act (HIPAA) which requires covered entities to adopt administrative and technical safeguards in order to protect the privacy of personal medical information. Financial software systems in the USA must comply with the Sarbanes-Oxley act of 2002 (SOX), which establishes wide ranging standards for all U.S. public company boards, management, and public accounting firms. Furthermore, safety critical systems often have to satisfy a staggeringly large number of regulatory codes impacting both software and hardware components. These regulations impact almost every part of the system including its electrical, mechanical, operational, and software components.

Current practice requires systems and software engineers to identify the relevant set of regulatory codes and to establish traceability between those regulations and system level requirements. Traceability is defined by the Center of Excellence for Software Traceability (CoEST) as the ability to link requirements back to contributing sources such as stakeholders' rationales, hazards to be mitigated, and regulatory codes, and forward to corresponding design artifacts, code, and test cases (CoEST 2008). In fact, traceability is mandated across many safety-critical domains. The U.S. Food and Drug Administration (FDA) states that traceability must be used to verify that a software design implements all of its specified software requirements, that all aspects of the design are traceable to software requirements, and that all code is linked to established specifications and test procedures (FDS 2002). Similarly, the DO-178C standard (FAA 115C), which the USA Federal Aviation Administration (FAA) has established as the means of certifying that software aspects of airborne systems comply with airworthiness requirements, specifies a very detailed set of traceability requirements.

In large Systems Engineering projects, compliance can be extremely costly and time consuming. Berenbach et al. (2010) described a 30,000 requirement project with 300 different sets of relevant regulatory codes. The traceability costs for this compliance effort were estimated by the project's requirements specialist at over USD$1.6 Million. To establish traceability, engineers are often forced to manually review hundreds, or even thousands of pages of regulatory codes in order to identify relevant sections, and then to painstakingly trace individual regulations to product level requirements or implemented code. Prior work with our industrial collaborators has shown the viability of using automated techniques to identify relevant sections of the regulations (Berenbach et al. 2010) but the cost of establishing and validating individual trace links is still extremely high. As a result, several industrial case studies have reported that even in safety-critical systems, engineers often fail to deliver a complete, and accurate set of trace links (Rempel et al. 2015; Mäder et al. 2013; Gotel and Finkelstein 1994; 1997; Ramesh and Jarke 2001).

1.1 Current Solutions

To address these problems, researchers have investigated the use of information retrieval techniques, including the Vector Space Model, Latent Semantic Indexing, and probabilistic

networks (Antoniol et al. 2000; Cleland-Huang et al. 2007; Hayes et al. 2004; Marcus and Maletic 2000), to dynamically generate trace links between various types of artifacts. Most prior work has focused on tracing documentation to code (Antoniol et al. 2000, 2002; Mirakhorli and Cleland-Huang 2016) or design specifications to code (Sultanov et al. 2011), with limited attention paid to tracing between regulations and requirements (Guo et al. 2013; Guo et al. 2014). The effectiveness of automated methods has varied quite widely, depending upon the individual datasets (Lohar et al. 2013; Shin et al. 2015; Sultanov et al. 2011; Mahmoud and Niu 2015), with recall reported between 50 and 95 % and precision sometimes in the single digits for large industrial datasets.

1.2 Proposed Solution

In this paper we focus on the non-trivial problem of automating traceability between regulatory codes and system level requirements. This represents a particularly compelling and challenging problem because industries are often required by certifying bodies to demonstrate compliance against regulations; however, the sheer number of regulations means that a fully manual tracing effort can be time-consuming and costly. Unfortunately, given the significant disparity which often exists between the terminology used in regulatory codes and that used in requirements (Berenbach et al. 2010) standard Information Retrieval approaches often fail to deliver a relatively complete and accurate set of trace links. In this paper, we tackle this term-mismatch problem, with the specific goal of improving the accuracy of trace links generated between regulations and requirements.

We propose and evaluate three different techniques for addressing the term-mismatch problem based upon classification, web-mining, and ontology respectively. The three approaches require different amounts of upfront human effort and are applicable in different contexts.

- The **classification** technique uses a pre-existing, human-validated trace matrix to train a classifier to trace specific regulatory codes. The classifier described in this paper is an extension of our previously published paper entitled *A Machine Learning Approach for Tracing Regulatory Codes to Product Specific Requirements* (Cleland-Huang et al. 2010). Trace-by-Classification works particularly well in cases where a sufficiently large training set of examples is available, and when the regulations trace to a relatively homogeneous set of requirements.
- The **web-mining** technique utilizes information retrieval to mine terms and phrases from domain specific documents in order to replace or augment the original trace query. In general, this approach does not perform as well as the classifier, however, it has the advantage of not requiring a training set of linked requirements and can therefore be used when insufficient training data is available. The web-mining technique was initially presented in Cleland-Huang et al. (2010) and described and evaluated in depth in a subsequent paper entitled *Towards Mining Replacement Queries for Hard-to-Retrieve Traces* (Gibiec et al. 2010). The second paper provided a more rigorous analysis of the approach, explored various parameterizations, and analyzed results with respect to the kinds of hard-to-retrieve trace queries that are often found in regulatory codes. In this journal paper, we extend our analysis of both the classifier and web-mining techniques, reporting their effectiveness across ten individual projects.
- The **ontology** technique captures domain terminology and concepts (Gruber 1993) in an ontology and then uses this knowledge to dynamically augment the text in the query during the tracing process. While researchers have previously proposed the

use of ontology for such purposes (Kof et al. 2010), to the best of our knowledge the work we present here represents the first concrete study that demonstrates its efficacy for improving trace accuracy. The examples we provide in this paper utilize an ontology that was constructed manually with semi-automated support (Guo 2016). We propose and evaluate two different algorithms for integrating ontology into the tracing process.

Our experimental results show that the classification approach returned the best results when sufficient training data was available. The ontology solution also showed improvements, but required non-trivial effort to create a suitable ontology for a project. Finally, the web-mining technique showed improvements in only a subset of queries. As discussed throughout the remainder of the paper the three query augmentation techniques offer trade-offs in terms of performance, cost and effort, and usage viability within each project context.

2 Data Sets

In this paper, the three approaches are evaluated against the technical safeguards of the USA's Health Insurance Portability and Accountability Act (HIPAA) of 1996 [HIPAA], with a focus on security safeguards. These safeguards, shown in Fig. 1, describe mandatory features such as access control, audit controls, authentication, and transmission security. The HIPAA security safeguards are traced against the software requirements specifications (SRS) of ten electronic healthcare systems. During the summer of 2010 we hired an MS Information Systems student with prior industry experience to retrieve datasets of

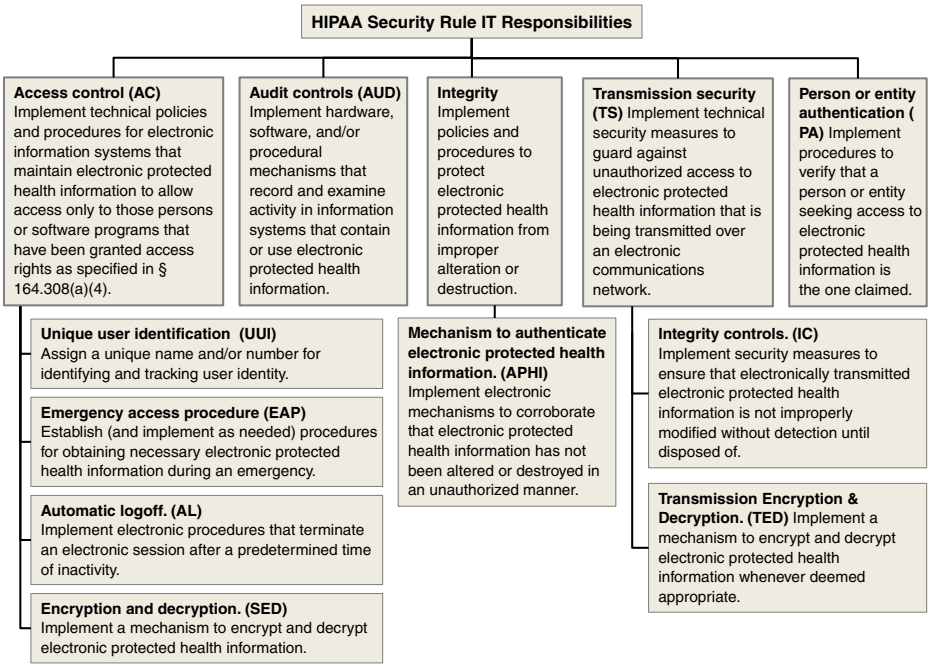


Fig. 1 HIPAA security rules and responsibilities

health-care related requirements. He identified a set of 10 systems in the form of open source products, IT healthcare standards, requirements exemplars, and feature descriptions for commercial products. Information for each of these projects, and their relevant source documents are summarized in Table 1. In cases where the requirements were extracted from product-level documentation or from online forums the original language and text was retained in order to ensure that no additional terminology was added. Table 1 also shows the number of requirements that were considered relevant for each of the HIPAA regulations. Two members of the team (including the original data retriever) separately reviewed all of the requirements, identified ones that were relevant to each HIPAA regulation, and constructed a trace matrix accordingly. Any discrepancies were resolved through discussion between four members of our research team at the time. The retrieval of the requirements, and subsequent construction of trace matrices, was therefore performed internally

Table 1 Healthcare related datasets used in experiments showing total number of requirements and number of HIPAA related requirements per type

Description	All	AC	AUD	AL	EAP	PA	SED	TED	TS	IC	UUI
Care2x: An open source Hospital Info. System (HIS). Care2x User Manual and Forum. http://www.care2x.org	44	1	1	1	0	1	1	1	0	0	0
CCHIT: Requirements for the Certification Commission for Healthcare Technology http://www.cchit.org	1064	17	33	1	1	12	2	2	2	5	3
ClearHealth: Open source HER. ClearHealth Web Site http://www.clear-health.com	44	1	4	1	0	0	1	1	0	2	1
Physician: Electronic exchange of info between clinicians. Use Cases. hmss.org/content/files/CTC_use_Case.pdf	147	7	2	0	2	0	0	0	1	3	0
iTrust: Open source HER. Use Cases. http://agile.csc.ncsu.edu/itrust/wiki/doku.php	184	3	35	1	0	6	0	0	0	0	2
Trial Implementations: National Coord. for Health Info. technology (HIT). Use Cases. http://healthit.hhs.gov	100	6	0	0	0	13	0	0	2	4	2
PatientOS: Open source healthcare info. System. PatientOS Web Site http://www.patientos.org	90	1	2	3	1	0	3	1	1	0	1
PracticeOne: A Suite of healthcare info. Systems. Practice One Website http://www.practiceone.com	34	3	1	0	0	1	0	0	1	1	0
Lauesen: Sample EHR requirements. (link no longer available)	66	11	0	1	0	5	0	0	0	3	1
WorldVistA: Veteran Administrations EHR VisA Manuals, EHR Application Features http:/worldvista.org	116	6	2	2	0	4	0	0	0	0	1
Total counts	1889	54	86	10	4	42	7	5	7	18	11

by authors of the original paper on trace link classification (Cleland-Huang et al. 2010); however, the datasets have since been utilized and validated by an external researcher from CalPoly. Our datasets are released to the PROMISE data repository (*upon acceptance of this paper).

The process of finding health-care related requirements specifications, preparing the datasets in a standard format, and constructing links took a total of 125 h. For illustrative purposes, the traceability matrix for the Automatic Logoff (AL) regulation is depicted in Table 2. It shows that a total of ten AL requirements were found across seven of the ten projects.

3 The Term Mismatch Problem

Traceability establishes *links* between *source* and *target* artifacts—for example, between requirements and source code, or between regulatory codes and requirements. A *query* is formulated using the text of a source artifact and the traceability technique is used to retrieve relevant *documents* from the set of target artifacts. In this paper, we trace HIPPA regulatory codes against Healthcare requirements. Therefore, HIPPA regulations are used as queries and requirements as target documents. We refer to a requirement that is related to regulation q as a *q type requirement*.

An analysis of four different datasets (i.e. CM1-NASA (Hayes et al. 2006), GasCodes (Berenbach et al. 2010), AREMA (Berenbach et al. 2010), and HIPAA) showed that in each case, 5–13 % of the targeted trace links exhibited significant term mismatches leading to underperformance.

A term mismatch occurs when language in the target document neither matches the language of the source document nor matches project level synonyms defined in a project glossary or standard thesaurus (Zou et al. 2010). In fact, this is such a common occurrence, that traceability tools, such as Poirot (Cleland-Huang et al. 2005), incorporate features which allow humans to manually add search terms and to filter out unwanted terms. This is illustrated in Fig. 2, which depicts a HIPAA regulatory code related to access control, traced

Table 2 Requirements associated with the HIPAA regulation for automated logoff

Dataset	Requirement
ClearHealth	System has a timer allowing automatic logoff
Care2x	System will implement session time outs and use cookies to terminate an electronic session
Lauesen	The system must time out if the user has been away from the system for some time
iTrust	Electronic session must terminate after a pre-determined period of inactivity. Administrator must be able to specify this period
WorldVistA	The system shall timeout after a period of inactivity
WorldVistA	The system shall ask the user if the user wants to continue using the system before timing out
PatientOS	Automatic timeout can be specified by location
PatientOS	Automatic timeout can be specified by role
PatientOS	When the system timeouts, the user is returned to the login page to sign in again
CCHIT	The system upon detection of inactivity of an interactive session shall prevent further viewing and access to the system by that session by terminating the session, or by initiating a session lock that remains in effect until the user reestablish-es access using appropriate identification and authentication procedures. The inactivity timeout shall be configurable

to the CCHIT requirements. Requirements are ranked and displayed according to the likelihood of their relevance (Zou et al. 2010). In this particular example, the user has added an additional term "authorize", and has filtered out several terms that she felt were causing imprecision in the results. The query was modified accordingly and the trace algorithm rerun to generate an updated list of potentially relevant requirements. These are depicted on the left side of Fig. 2.

Although this type of modification has been shown to improve the recall and precision of certain trace queries (Shin and Cleland-Huang 2012), its effectiveness is dependent upon the domain knowledge of the person performing the trace, and also upon their willingness to manually modify the query. Unfortunately, prior studies have demonstrated that users lose confidence in a traceability tool that returns inconsistent or imprecise results, and that human errors are introduced when human analysts are asked to evaluate a long list of candidate links (Hayes et al. 2004; Cuddeback et al. 2010). The approaches we describe in this paper therefore represent possible solutions for automating the query-modification process either explicitly prior to running the query or intrinsically as the query is being processed.

4 Evaluation of a Baseline Approach

We used the Vector Space Model (VSM) proposed by Salton and McGill (1986) and Salton et al. (1975) to create a baseline against which to compare our techniques. In a series of prior experiments we used a Genetic Algorithm to search for the best tracing configuration for six different datasets of various sizes and domains and found VSM to be the winner in five out of six cases (Lohar et al. 2013). We therefore deemed VSM an appropriate baseline for use in our current work.

VSM represents each *query* q (i.e. the source artifact) as a vector in multidimensional space $q = (w_{1,q}, w_{2,q}, ...w_{t,q})$ in which each term is represented by its own dimension and weighted according to its importance. Documents (i.e. target artifacts) are similarly

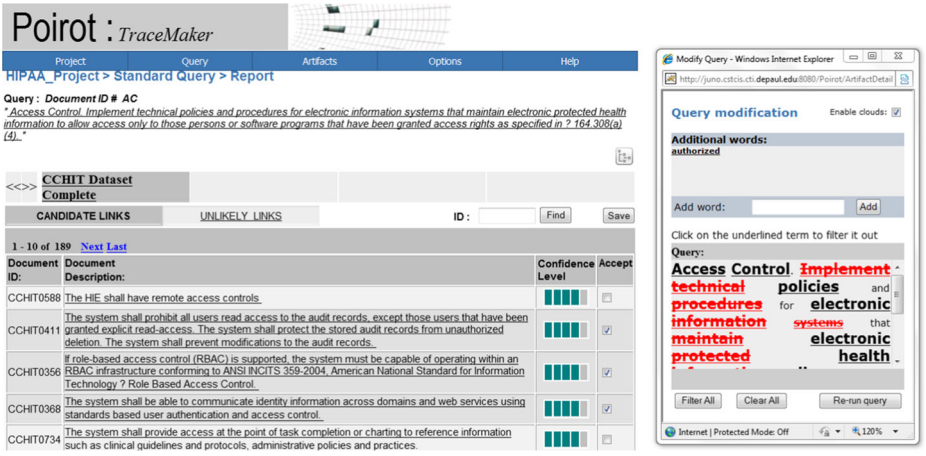


Fig. 2 HIPAA access control regulation query manually modified in the Poirot tracing tool

represented $d_j = (w_{1,j}, w_{2,j}, ...w_{t,j})$. The similarity between query q and document d_j is computed as the cosine of the angle between the two vectors as follows:

$$sim(d_j, q) = cos\theta = \frac{d_j \cdot q}{\parallel d_j \parallel \parallel q \parallel} = \frac{\sum_{j=1}^{N} w_{i,j} w_{i,q}}{\sqrt{w_{i,j}^2 w_{i,j}} \sqrt{w_{i,j}^2 w_{i,q}}} \quad (1)$$

Term t in document d is typically weighted using the *term frequency, inverse document frequency* (tf-idf) computed as follows:

$$w_{t,d} = tf_{t,d} \cdot log \frac{|D|}{|\{d' \in D | t \in d'\}|} \quad (2)$$

where $tf_{t,d}$ represents the term frequency of term t in document d, $|D|$ is the total number of documents in the set, and $|\{d' \in D | t \in d'\}|$ represents the number of documents that contain term t.

In preparation for use, each requirement and each regulatory code is preprocessed to remove *stop words*, i.e. commonly occuring words such as "system" and "shall". Next, because words with similar semantic interpretations often have morphological variants, the remaining words are stemmed to a common root form using the Porter's stemming algorithm (Porter 1980). Conceptually, the terms from the regulatory code serve as the trace query q while the terms in the requirements serve as the documents.

4.1 Trace Retrieval Metrics

Information retrieval results are typically evaluated using *recall* and *precision* metrics (Salton 1989), where recall measures the fraction of relevant links that are retrieved and precision measures the fraction of retrieved links that are relevant. For experimental purposes, a threshold score is established such that all links above or at the threshold are retrieved and all links below the threshold are rejected. In order to compare recall and precision results across experiments the F-Measure, which computes the harmonic mean of recall and precision is often adopted. We use a variant of the F-Measure, known as the F2-Measure, which weights recall values more highly than precision. This measure is commonly used to evaluate experiments in the traceability domain where recall is valued more highly than precision. The F2-Measure is computed as follows:

$$F2\ Measure = \frac{5 \times Precision \times Recall}{(4 \times Precision) + Recall} \quad (3)$$

Finally, we adopt another well-used metric of Mean Average Precision (MAP) which evaluates the extent to which relevant links are listed near the top of a ranked list as opposed to lower down the list. First, Average precision (AP) is computed as:

$$AveragePrecision = \frac{\sum_{rank=1}^{|Retrieved|} (Precision(rank) \times relevant(rank))}{|RelevantLinks|} \quad (4)$$

where $|Retrieved|$ represents the number of retrieved links, *rank* is the position of the requirement in the ordered set of candidate traceability links, $relevant()$ is a binary function assigned 1 if the rank is relevant and 0 otherwise, and $Precision(rank)$ is the precision computed after truncating the list immediately below that ranked position. Mean Average Precision (MAP) is then computed as the mean AP across a set of queries. MAP is a very meaningful metric in the traceability domain as it has the ability to capture a "perfect" trace in which all correct links are listed at the very top. These metrics are used throughout the remainder of the paper to document and compare results from each of the experiments.

4.2 Baseline Results

VSM was used to generate trace links from the ten targeted HIPAA regulations to requirements in each of the 10 patient healthcare systems. Trace queries were formulated directly from the title and description of each HIPAA security regulation as shown in Fig. 1. Recall, precision, and F2-Measure metrics were computed individually for each of the HIPAA regulations by establishing threshold values that optimized the F2-Measure. The decision to establish threshold values in this way was applied consistently for all experiments reported in this paper. The MAP metric was computed to include all targeted links, which means that 100 % recall is assumed. For cases in which the probability of the targeted link was predicted at 0.0 (i.e. there were no shared terms between the regulatory code and the related requirement), we computed the position of the requirement in the ranked list as the average position of all requirements returning probability scores of 0.0. These cases correctly reduced the average precision quite significantly. Utilizing average precision in this way, therefore effectively takes both recall and precision into account. The results from this baseline experiment are reported in Table 3.

Top results were observed for *Emergency Access Procedure* (EAP), *Storage Encryption and Decription* (SED), and *Automatic Logoff* (AL). For example, AL achieved recall of 0.881, precision of 0.651, and MAP of 0.634. However, on the other end of the spectrum, HIPAA regulations such as *Integrity Control* (IC), *Personal Authentication* (PA), and *Transmission Security* (TS) all returned MAP scores lower than 0.4 which is unacceptable. A strong correlation (0.94 using the Pearson Product-Moment Correlation Coefficient) exists between F2-Measure and MAP scores, therefore we only discuss MAP here. In general, VSM is only effective when the terms co-occur across both the query (regulations) and documents (requirements). The overall lack-luster results indicated that the term-mismatch problem hindered the effectiveness of trace link generation when VSM was used.

In the following sections we describe the three query augmentation techniques. All experiments have been rerun for purposes of this paper to ensure consistency in the way metrics are computed and reported. Furthermore, several algorithms have been reconfigured in order to improve their performance. Such changes are fully described in this paper. Due to rerunning experiments, there are some differences in results from those reported in our original work (Cleland-Huang et al. 2010; Gibiec et al. 2010).

Table 3 Accuracy of trace links generated using the VSM baseline

HIPAA regulation	Recall	Precision	F2	MAP
Access control (AC)	0.742	0.375	0.459	0.336
Automatic logoff (AL)	0.881	0.651	0.800	0.634
Audit control (AUD)	0.888	0.295	0.535	0.264
Emergency access procedure (EAP)	1.000	0.576	0.720	0.633
Integrity control (IC)	0.664	0.149	0.309	0.144
Personal authentication (PA)	0.707	0.138	0.323	0.162
Storage encryption & decryption (SED)	1.000	0.875	0.958	0.875
Transmission encryption & decryption (TED)	1.000	0.439	0.715	0.536
Transmission security (TS)	1.000	0.242	0.576	0.337
Unique user ID (UUI)	0.929	0.518	0.686	0.526
Average	0.881	0.426	0.608	0.445

5 Classification Technique

Machine learning methods are particularly appealing for tracing regulatory codes, such as HIPAA safeguards, because the upfront effort of training a classifier can be potentially recouped when those same regulations are applied across additional projects. Although there are many different classification algorithms, we adopted one that we had previously developed for classifying non-functional requirements (NFRs) (Cleland-Huang et al. 2007; 2006). We have used the same algorithm to classify source code snippets which implement architectural tactics such as *heart-beat* or *resource pooling* (Mirakhorli et al. 2012). Our study showed that the NFR Classifier outperformed standard classification techniques including the Näive Bayes classifier, decision tree (J48), feature subset selection (FSS), correlation-based feature subset selection (CFS), and various combinations of the above (Mirakhorli and Cleland-Huang 2016).

5.1 Training the Classifier

The classification process requires a training set of regulatory codes, software requirements, and their associated trace links. All data is prepared using the preprocessing techniques applied to VSM.

During the training phase, each term in the requirements specification is weighted according to the degree to which it represents a HIPAA regulation. We refer to terms that occur freqently in requirements related to specific regulations as *indicator terms*. For example, the term *send* is frequently found in requirements related to *transmission* but is far less likely to occur in requirements related to *automatic logoff*. It is therefore assigned a relatively strong indicator weighting with respect to *Transmission Encryption and Decryption*.

Indicator term weightings for regulation q are computed by considering the set S_q of all requirements in the training set that are related to regulation q. The cardinality of S_q is defined as N_q and each term t is assigned a weight score $W_q(t)$ that depicts the extent to which the term identifies a requirement of type q. The weight score $W_q(t)$ corresponds to the probability that a particular term t identifies a requirement as being associated to regulation q based on the standard Information Retrieval assumption that terms indicating relevance for a certain targeted regulation must be present in the document to be classified. $W_q(t)$ is computed by considering three different factors. The first factor $\frac{freq(t, d_q)}{|d_q|}$ computes the frequency at which term t occurs in requirement d_q. It is computed individually for each requirement in S_q. The second factor $\frac{N_q(t)}{N(t)}$ computes the fraction of q type requirements that contain term t, while the third factor, $\frac{NP_q(t)}{NP_q}$ computes the fraction of projects which include requirements related to regulation q, which also contain term t. This final factor reduces the impact of project-specific indicator terms. The formula is shown below:

$$W_q(t) = \frac{1}{N_q} \sum_{d_q \in S_q} \frac{freq(d_q, t)}{|d_q|} * \frac{N_q(t)}{N(t)} * \frac{NP_q(t)}{NP_q} \tag{5}$$

where d_q is a q type requirement document in the training set; N_q is the number of regulations; $N_q(t)$ is the number of q type training requirement documents with term t; $N(t)$ is

the number of all type training requirement documents with term t; $NP_q(t)$ is the number of projects that include q type requirements with term t; NP_q is the number of projects that include q type requirements. A weighting $W_q(t)$ is computed for each term t with respect to q, and terms are then ranked by decreasing order according to $W_q(t)$. In our previous work Cleland-Huang et al. (2010), we utilized top ten indicator terms for each regulation; however, additional experiments performed for this paper showed that utilizing all indicator terms returned better results. Therefore all indicator terms have been used throughout the experiments reported here.

5.2 Using the Classifier

Once indicator term weights have been computed, the classifier computes a score $Pr_q(d)$ which represents the probability that a given requirement d is associated with the regulation q. The underlying assumption is that type q requirements are more likely to contain indicator terms for that type.

Let I_q be the set of indicator terms for regulation q identified during the training phase using Formula (5). The classification score that a requirement document d belongs to regulation R is then defined as follows:

$$Pr_q(d) = \frac{\sum_{t \in d \cap I_q} W_q(t)}{\sum_{t \in I_q} W_q(t)} \tag{6}$$

where the numerator is computed as the sum of the term weights of all indicator terms for regulation q that are contained in d, and the denominator is the sum of the term weights for all type q indicator terms. In this way, the classifier will assign a higher score $Pr_q(d)$ to a requirement document d that contains several strong indicator terms for q.

5.3 Evaluating the Classifier

Given the fact that we had only 10 distinct datasets, we adopted a ten-fold cross validation experimental design. In each iteration of the experiment, nine projects were used to train the classifier. The trained classifier was then used to classify requirements in the remaining project. The process was repeated until each project had been tested (i.e. classified) one time. Based on initial experimentation (Cleland-Huang et al. 2010), we adopted a multi-classification approach in which all requirements that scored higher than a certain threshold value were classified as relevant to that regulation. An individual requirement could therefore be assigned to more than one HIPAA regulation. As with the previous experiment, threshold values were set individually for each query so as to maximize F2-measure values.

We report classifier results for each HIPAA regulation in Table 4. Figure 3 compares results against the VSM baseline. Notably, the classifier improved results in eight out of ten cases. In six of the ten cases (AC, AL, AUD, PA, TED, TS) the classifier approach improved results, in two cases it was closely equivalent (EAP, SED), and in two cases (IC, UUI) it performed worse than the baseline VSM approach. Surprisingly, we found that even in the case of relatively small training sets, the classifier performed quite well. The Wilcoxon-Signed rank test with continuity correction showed a significant improvement when comparing the classifier results to VSM (p-value = 0.0002842). This result improves on our previous observations. We attribute the improved performance to our use of all indicator terms to represent each regulation instead of the top 15 used in our previous work (Cleland-Huang et al. 2010).

Table 4 Accuracy of trace links generated using the classifier

HIPAA regulation	Recall	Precision	F2	MAP
Access control (AC)	0.855	0.490	0.663	0.499
Automatic logoff (AL)	1.000	0.774	0.877	0.839
Audit trail (AUD)	0.962	0.641	0.798	0.718
Emergency access procedure (EAP)	1.000	0.666	0.874	0.638
Integrity controls (IC)	0.561	0.153	0.273	0.112
Personal authentication (PA)	0.854	0.675	0.798	0.737
Storage encryption & decryption (SED)	1.000	0.750	0.916	0.854
Transmission encryption & decryption (TED)	1.000	0.916	0.977	0.958
Transmission security (TS)	0.800	0.523	0.660	0.481
Unique user ID (UUI)	0.928	0.399	0.563	0.388

To understand why the classifier performed well in the majority of cases, we examine the original query terms alongside two sets of indicator terms in Table 5. The first column lists original query terms, while the next two columns list the top 15 ranked indicator terms learned when CCHIT (column 2) and Trial Implementations (column 3) were set aside for testing. Column 2 indicator terms would therefore be used to test the CCHIT dataset and Column 3 for Trial Implementation etc. On average each HIPAA regulation has 111 indicator terms, although many are weighted with very low scores. The terms learned from the training set provide a far richer explanation of the HIPAA regulations, than the original query. Using the modified query can therefore result in significant improvements in MAP scores.

We analyze the two cases in which the classifier performed worse than VSM (i.e. IC and UUI). In the case of Integrity Constraints (IC) there are 18 requirements dispersed across six datasets. However, the requirements are very diverse and cover topics such as integrity of patient data, retention and purging of data, and sending acknowledgements once transmitted data is correctly received. In the second case of Unique User ID (UUI),

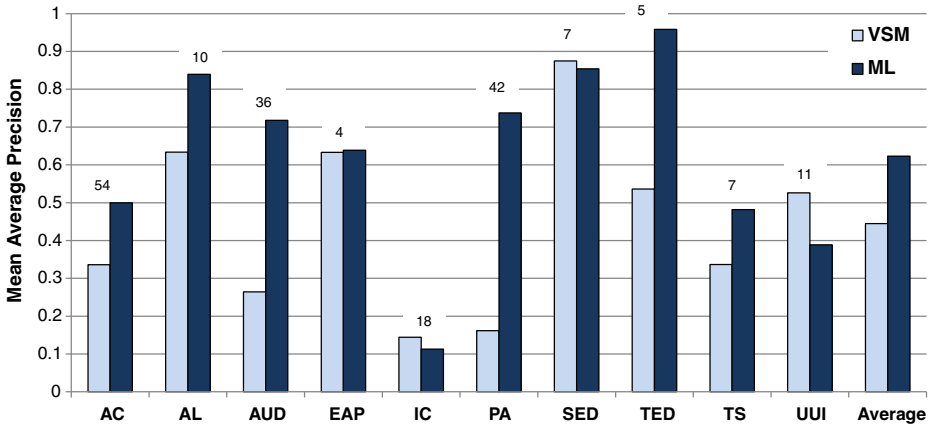


Fig. 3 VSM versus classification (ML). Numbers above each column represent the count of each requirement type

Table 5 Key terms in original vs. machine learned modified queries

Reg	Original query	Indicator terms (CHHIT as test)	Indicator terms (trial implementations as test)
AC	access allow control electron grant health implement inform maintain person polici procedure program protect right softwar specifi technic	access role user provid secur assign privileg rights controls allow author permiss enforc right consum	access role user secur allow assign privileg rights controls author permiss restrict right provid hospital roles
AL	automat electron implement inact logoff predetermin procedur session termin time	timeout session specifi automat period termin inactivity out time user timer awai logoff return location electron	timeout inactivity session termin automat period out awai logoff timer specifi user location again timeouts time
AUD	activ audit contain control electron examin hardwar health implement inform mechan procedur protect record software	log record audit perform transact everytim interact events user patient access support actions track chang	log record perform audit transact everytim user patient support creat authent time provid access note
EAP	access electron emerg establish health implement inform necessari need obtain procedur protect	emerg phi emergency situations popul workflow access public individu offici individual creat situat electron custom	situations emerg workflow emergency popul access restricted individual offici glass break appropri individu public health
IC	control detect dispos electron ensur health implement improperli inform integr measur modif	format consist data concurr verac integrity deleted problems ensur warn found rbac model standard source	integrity data consist format deleted validity problems concurr doubt metadata record model compliant emploi valid
PA	access authent claim electron entiti health implement inform person procedur protect seek verifi	password authent author user passwords method identifi users provid reset access allow support identification protect	password authent user used author passwords reset allow support method automatically sensit identifi access protect
SED	decrypt electron encrypt health implement inform mechan protect	encrypt databas encrypted tls individu field sl ssl us requir data	encrypt encrypted databas standards field individu sl tls ssl aes 3des us hashing encryption pda
TED	appropri deem electron encrypt health implement inform mechan protect	ssl sl encrypt hash server support modif prevent option everyth offer transit transmit client us commun	ssl sl encrypt hash open server everyth transit offer modif support client option transmit tls internet
TS	access commun electron guard health implement inform measure network protrect secur techni transmiss	transmission secur secreci sender hitsp maintain content deliveri assum construct electron s privaci transport delivery	secur messag ensur electron browser html delivery deliveri confirm perimet ehrs recipient networks web interfac
UUI	assign ident identif identifi number track uniqu user	uniqu number identifi correctli identif systems medic user usernam requested authentication patients password protocol enforc method	uniqu number identif identifi user hies ldap usernam lightweight ensur us enforc password authentication patients

The unstemmed version of the original query is shown in Fig. 1

the requirements integrate concepts related to unique IDs with a broad range of additional concepts, again leading to diversity of requirements. This analysis suggests that the Classification technique is most effective when requirements provided in the training set are

relatively homogeneous. In both the IC and UUI regulations, the HIPAA guidelines could benefit from further refinement.

Acquiring or developing training sets can be time consuming and is not always feasible; however, our results suggest that even relatively small training sets can be effective. Furthermore, constructing a training set might not be that difficult for an organization that builds multiple products that all comply to similar regulations. Another alluring option is for certifying bodies to provide trained classifiers which organizations could use in order to check their systems for compliance.

6 Web-Based Query Augmentation

Our second approach uses web-mining to augment the terms in a trace query. The process is illustrated in Fig. 4. A regulatory code is passed to one or more standard search engines, in order to search for, and retrieve, relevant documents. Any documents which are primarily graphical in nature or composed mainly of advertisements are automatically discarded.

Each of the retrieved documents is partitioned into sections by splitting it into overlapping chunks of length *chunkLength*. A parameter, *chunkOffset* determines the distance in characters between the starting point of the previous chunk and the starting point of the current chunk. Creating overlapping chunks ensures that unfortunate partitioning decisions do not prevent the algorithm from recognizing the most relevant section of text. We evaluate the use of different chunking parameters in Section 6.4 of this paper.

For each document, we use VSM to compute the similarity between the chunk and the original HIPAA regulation in order to identify the chunk which exhibits the highest similarity. This most relevant chunk is then analyzed further using an approach we previously developed to extract project glossary terms and phrases from requirements specifications (Zou et al. 2010). We first parse the text of the relevant chunk to extract nouns and noun phrases. Our prior work (Zou et al. 2010) indicated that nouns and noun phrases were highly representative of the content of a trace query. We therefore focus on these parts of speech for purposes of augmenting the query. We used QTag, a part-of-speech (POS) tagger to identify nouns and noun phrases (Tufis and Mason 1998). QTag identifies the syntactic category of each token in the text and outputs a series of POS tags that represent grammatical classes of words, such as nouns, verbs and adjectives for each token in the input text. These tags are then used to extract both nouns and noun phrases (from now on referred to as 'phrases'). A

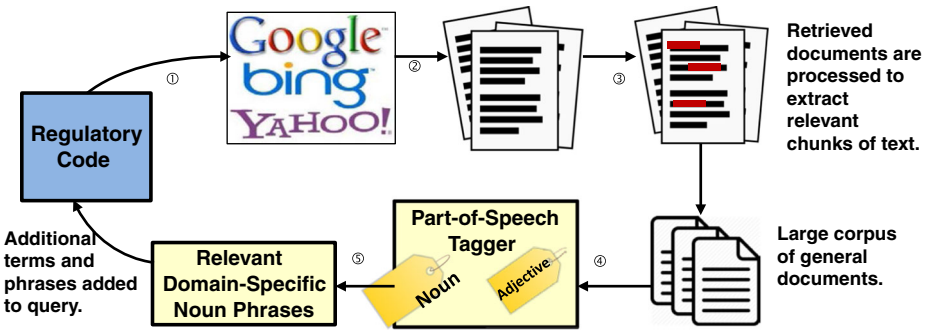


Fig. 4 Query modification technique

typical trace query produces between 5,000 and 10,000 nouns and noun phrases. Examples taken from HIPAA's audit control regulation are depicted in Table 6.

Finally, we compare the frequency with which nouns and noun phrases appear in requirements related to each HIPAA regulation versus their occurrence in a general domain corpus. For purposes of these experiments, we constructed a general domain corpus from a collection of over 50 e-books and other online documents that covered topics as broad as science fiction, business, nature, self-help, and even romance. These documents allow us to compute *Domain Term Frequency (DTF)* and *Domain Specificity (DS)* metrics which are used to rank and filter terms and phrases for use in the augmented trace query. The metrics are defined as follows:

- **Domain Term Frequency (DTF)** computes the normalized term frequency for term t across multiple documents as follows: $DTF(t) = \sum_{d \in D}(freq(t,d)/|d|)$ where $freq(t,d)$ is the total number of occurrences of term t in a given domain-specific document d, and $|d|$ is the length of that document expressed as the total number of terms in d. Such normalized occurrences from all retrieved domain-specific documents D are added together.
- **Domain Specificity (DS)** measures the extent to which a term or term phrase is specific to documents in the targeted domain in comparison to the frequency of its more general use. The domain specificity $DS(d,t)$ of term t in document d, is computed by comparing the relative frequency at which the term appears within a domain-specific document versus its relative frequency in a general corpus of documents [(Zou et al. 2010)]. DS is calculated as follows:

$$DS(t,d) = ln\left[\frac{freq(t,d)}{\sum_{u \in d} freq(u,d)} / \frac{freq(t,G)}{\sum_{v \in G} freq(v,G)}\right] \tag{7}$$

  where the first element $(freq(t,d))/(\sum_{u \in d} freq(u,d))$ is the normalized number of occurrences of term t in the domain-specific document d, and the second element is the normalized number of occurrences of t in the general corpus of documents. Domain specificity (DS) is then calculated as the average value of all domain specificities from each document from the collection:

$$DS(t) = \frac{1}{|D|} \sum_{d \in D} DS(t,d) \tag{8}$$

Table 6 Sample terms generated from the HIPAA audit control regulation

Term	DS	CG	DTF	Term	DS	CG	DTF
audit	67.2	0.8	5.1	work	14.7	0.2	0.5
control	9.0	0.9	5.1	board	63.7	0.2	0.5
manag	8.5	0.7	2.0	respons	8.2	0.3	0.5
secur	28.2	0.5	1.7	technolog	20.4	0.3	0.5
report	5.5	0.6	1.4	access	3.9	0.2	0.5
risk	473.0	0.5	1.4	record	6.7	0.2	0.5
system	16.7	0.6	1.3	time	1.2	0.3	0.5
process	5.1	0.5	1.3	audit control	1000.0	0.2	0.5
auditor	88.5	0.4	1.2	committe	1000.0	0.2	0.5

In cases where a term is not found in our general corpus, we consider it highly domain specific and assign it the maximum DS value of 1000.0.

DS and DTF metrics are computed for each phrase. Phrases exhibiting domain specificity below a predefined threshold are filtered out and the remaining phrases are ranked in descending order of term frequency. In the web-mining approach, the original query is entirely replaced with the set of terms and phrases that score above the threshold in terms of DS and DTF scores.

6.1 Applying Web-Augmentation to HIPAA Regulations

A series of experiments, originally reported in Gibiec et al. (2010), were conducted to evaluate and fine-tune the query augmentation process and related algorithms. To support these experiments we developed a Java based tool. For each HIPAA regulation it issued a Google, Bing, and Yahoo search. The retrieved documents were parsed, domain-specific phrases were identified, and the top terms or phrases exhibiting appropriate levels of term frequency (DTF) and domain specificity (DS) were selected. The tool outputs the set of reconstituted queries, which are passed to VSM in order to generate trace links. Examples are depicted in Table 7.

Table 7 Key terms in original vs. web-augmented modified queries (Note: The original query shown here is the preprocessed form of the HIPAA regulatory codes shown in Fig. 1)

Reg	Original query	Modified query
AC	access allow control electron grant health implement inform maintain person polici procedure program protect right softwar specifi technic	access control door reader permiss control system com role lock ident kei
AL	automat electron implement inact logoff predetermin procedur session termin time	logoff post shutdown com auto logon inact forum password login
AUD	activ audit contain control electron examin hardwar health implement inform mechan procedur protect record software	audit risk auditor procedur review complianc govern depart board audit control
EAP	access electron emerg establish health implement inform necessari need obtain procedur protect	emerg procedur health plan implement depart emerg access fire offic build
IC	control detect dispos electron ensur health implement improperli inform integr measur modif	integr
PA	access authent claim electron entiti health implement inform person procedur protect seek verifi	authent entiti health procedur ident implement certif identif password kei
SED	decrypt electron encrypt health implement inform mechan protect	encrypt kei algorithm password disk bit drive certif cipher decrypt
TED	appropri deem electron encrypt health implement inform mechan protect	encrypt decrypt kei algorithm password cipher string byte bit
TS	access commun electron guard health implement inform measure network protrect secur techni transmiss	secur encrypt transmiss kei health risk integr email data com
UUI	assign ident identif identifi number track uniqu user	identif password authent kei login mail card com

We configured the web-mining approach through conducting a series of experiments designed to evaluate different search engines, to determine the number of documents to retrieve, to configure chunk-size and chunk-offsets, and to establish the right threshold values for each metric in order to select the best domain-specific phrases.

6.2 Use of Different Search Engines

The first experiment compared results from using Bing, Google, and Yahoo search engines. In each case we retrieved 100 documents per query and partitioned the documents into chunks of 2000 characters. The previously described process was then applied. Results, which are reported in Table 8 indicate that the best performing search engine differed across the HIPAA regulatory codes. As a result, we used all three search engines in all other experiments reported in this paper.

6.3 Number of Documents Retrieved

The second experiment evaluated the impact of retrieving different numbers of documents. We evaluated the case of 10, 20, 40, 60 80 and 100 documents with each search engine. Of the retrieved documents we parsed html, doc (or docx), and pdf files) and additionally filtered out documents containing almost no text. Furthermore, because the Google Search Engine API only returns a maximum of 64 links in results sets of eight, Google results included 16, 24, 40, 64, 64, and 64 documents respectively. As a result the actual number of unique documents retrieved averaged 22, 45, 100, 160, 190, and 220 for each experiment. Results are depicted in Table 9. Individual results were influenced by the availability of high quality documents. As there was no clear winner we decided to use 10 documents because this reduced processing time.

6.4 Chunk Size

In our third experiment, we evaluated the impact of different chunk sizes on the trace query results. Previously stated default values were used, and chunk sizes of 200, 400, 800, 1000, 1600, and 3,200 characters were evaluated. Results for each HIPAA regulation are reported

Table 8 Recall, precision, and MAP scores achieved using different search engines

	Bing			Google			Yahoo			NGY (All)		
	Recall	Prec	MAP	Recall	Prec	MAP	Recall	Prec	MAP	Recall	Prec	MAP
AC	0.93	0.04	0.16	0.93	0.08	0.15	0.91	0.04	0.15	0.91	0.10	0.17
AL	0.10	0.33	0.04	0.10	1.00	0.11	0.40	0.33	0.28	0.40	0.37	0.29
AUD	0.89	0.05	0.10	0.91	0.05	0.10	0.90	0.05	0.12	0.90	0.05	0.10
EAP	1.00	0.06	0.30	1.00	0.13	0.29	1.00	0.12	0.15	1.00	0.09	0.19
IC	0.23	0.14	0.04	0.24	0.19	0.06	0.82	0.01	0.04	0.82	0.01	0.03
PA	0.59	0.09	0.16	0.55	0.06	0.06	0.57	0.06	0.06	0.57	0.07	0.11
SED	1.00	0.26	0.42	1.00	0.37	0.50	1.00	0.15	0.30	1.00	0.18	0.39
TED	0.80	0.11	0.10	0.80	0.10	0.10	0.08	0.11	0.09	0.80	0.11	0.10
TS	1.00	0.05	0.36	1.00	0.07	0.31	1.00	0.05	0.37	1.00	0.05	0.38
UUI	0.82	0.04	0.08	0.64	0.04	0.05	0.46	0.04	0.05	0.82	0.04	0.13

Table 9 Impact of retrieved document count on trace query results

	10 Results			20 Results			40 results			60 results			80 Results			100 results		
	Rec	Pr	MAP	Rec	Pr	MAP	Rec	Pr	MAP	Rec	Pr	MAP	Rec	Pr	MAP	Rec	Pr	MAP
AC	0.93	0.04	0.23	0.93	0.04	0.23	0.91	0.07	0.16	0.93	0.09	0.18	0.93	0.05	0.14	0.91	0.05	0.17
AL	0.10	1.00	0.11	0.10	1.00	0.11	0.10	1.00	0.11	0.20	0.01	0.11	0.40	0.40	0.30	0.40	0.36	0.29
AU	0.91	0.05	0.10	0.90	0.05	0.10	0.89	0.05	0.11	0.90	0.05	0.11	0.90	0.05	0.11	0.90	0.05	0.10
EA	1.00	0.08	0.26	1.00	0.09	0.12	1.00	0.10	0.20	1.00	0.11	0.25	1.00	0.09	0.19	1.00	0.09	0.19
IC	0.35	0.06	0.03	0.30	0.06	0.03	0.23	0.07	0.03	0.47	0.04	0.03	0.82	0.01	0.03	0.82	0.01	0.03
PA	0.57	0.07	0.08	0.60	0.08	0.17	0.57	0.07	0.07	0.57	0.06	0.07	0.55	0.06	0.07	0.57	0.06	0.07
SE	1.00	0.33	0.61	1.00	0.30	0.61	1.00	0.32	0.50	1.00	0.18	0.40	1.00	0.18	0.39	1.00	0.18	0.39
TE	1.00	0.08	0.09	0.80	0.05	0.04	0.80	0.10	0.06	0.80	0.12	0.10	0.80	0.11	0.10	0.80	0.11	0.10
TS	1.00	0.05	0.35	1.00	0.07	0.30	1.00	0.06	0.40	1.00	0.05	0.41	1.00	0.05	0.38	1.00	0.05	0.38
UI	0.45	0.03	0.16	1.00	0.02	0.11	0.82	0.04	0.10	0.55	0.03	0.11	0.64	0.03	0.08	0.82	0.04	0.11

Table 10 Metric results by chunk size

	Size: 200			Size: 400			Size: 800			Size: 1000			Size: 1600			Size: 3200		
	Rec	Pr	MAP	Rec	Pr	MAP	Rec	Pr	MAP	Rec	Pr	MAP	Rec	Pr	MAP	Rec	Pr	MAP
AC	0.91	0.05	0.19	0.91	0.04	0.17	0.93	0.04	0.14	0.91	0.04	0.17	0.93	0.04	0.15	0.93	0.05	0.16
AL	0.70	0.11	0.53	0.70	0.13	0.53	0.50	1.00	0.50	0.50	1.00	0.50	0.50	1.00	0.50	0.40	0.33	0.34
AU	0.90	0.05	0.14	0.90	0.05	0.11	0.90	0.05	0.10	0.90	0.05	0.10	0.90	0.05	0.11	0.90	0.05	0.05
EA	1.00	0.08	0.50	1.00	0.09	0.37	1.00	0.09	0.49	1.00	0.11	0.19	1.00	0.06	0.22	1.00	0.06	0.06
IC	0.41	0.04	0.05	0.59	0.03	0.04	0.35	0.04	0.04	0.35	0.05	0.03	0.24	0.07	0.03	0.35	0.04	0.04
PA	0.57	0.06	0.07	0.57	0.07	0.08	0.57	0.07	0.07	0.57	0.07	0.07	0.55	0.07	0.07	0.57	0.06	0.06
SE	1.00	0.32	0.64	1.00	0.29	0.55	1.00	0.23	0.30	1.00	0.21	0.42	1.00	0.18	0.39	1.00	0.19	0.19
TE	0.80	0.09	0.09	0.80	0.10	0.10	0.80	0.08	0.07	0.80	0.11	0.09	0.80	0.12	0.10	0.80	0.13	0.13
TS	1.00	0.03	0.30	1.00	0.03	0.16	1.00	0.04	0.25	1.00	0.04	0.30	1.00	0.11	0.40	1.00	0.06	0.06
UI	0.82	0.02	0.06	0.82	0.04	0.16	0.82	0.04	0.15	0.82	0.04	0.15	0.82	0.04	0.11	0.73	0.05	0.05

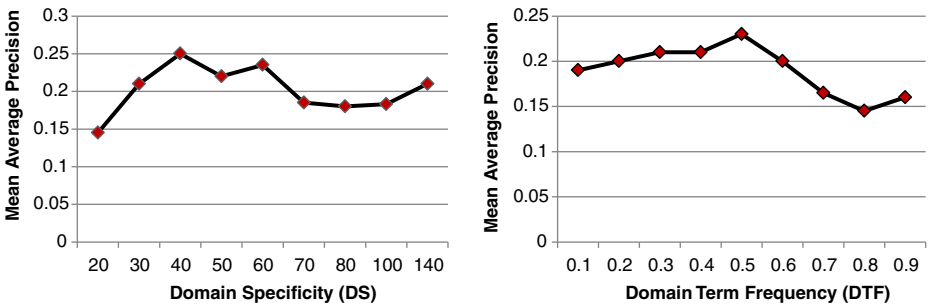


Fig. 5 The impact of domain specificity (DS) and domain term frequency (DTF) on mean average precision

in Table 10 and show that smaller chunk sizes were effective in six of the ten cases, larger chunks in two cases, and inconsistent results in the remaining cases. These results suggest that chunk size should not be fixed in length but should be customized according to the characteristics of each query and document.

6.5 Metric Calibration

Finally, we evaluated various combinations of threshold values DS and DTF metrics in order to filter the list of candidate terms and phrases generated by our web-mining approach. For this experiment, we used 100 documents per search engine, and a chunk size of 2000 characters with chunk offset of 200. We ran 512 experiments using diverse combinations of metric values as follows: DS (10–140), and DTF (0.1–0.9). Specific values within each range were selected by first testing combinations of low, medium, and high values for each metric, and then conducting a more detailed evaluating in areas of high-performing combinations. Results are reported in Fig. 5.

Table 11 reports top MAP scores from all 512 experiments. DS performed best with threshold scores ranging from 20 to 60. DTF performed best in the range of 0.1 and 0.4 with performance declining at higher values. Maximum MAP scores were achieved at threshold values of DS = 40 and DTF = 0.1 or 0.2. Furthermore, we observed that results were most sensitive to changes in DS. Setting it too high resulted in the exclusion of useful terms, while setting it too low resulted in the retention of overly general terms and phrases. For term frequency best results were achieved by setting the threshold to include less frequent terms which are representative of the domain.

6.6 Analysis

We report final results for our approach in Fig. 6. In six of the ten regulations (i.e. AC, AUD, EAP, IC, PA, and SED), Web-Mining improved accuracy of the MAP scores over

Table 11 Mean average precision for various combinations of metric values

Top scoring results					Sample of lower ranked results							
MAP	0.2602	0.2602	0.2585	0.2581	0.2567	0.2567	0.2502	0.2502	0.2265	0.1954	0.1807	0.1091
DS	40	40	40	40	40	40	40	40	50	15	50	80
DTF	0.1	0.2	0.3	0.4	0.4	0.4	0.29	0.29	0.25	0.5	0.2	0.9

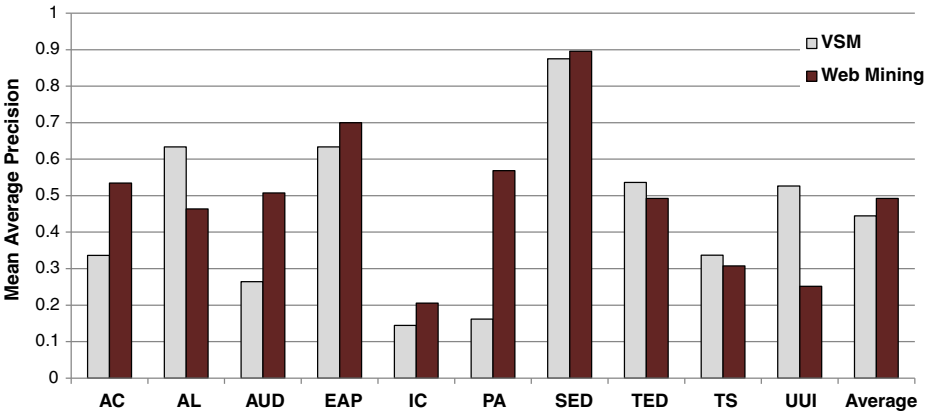


Fig. 6 VSM versus web-mining technique

the baseline VSM method. In four remaining cases (i.e. AL, TED, TS, and UUI) it did not. A small increase in average results was observed with VSM returning an average MAP of 0.445, compared to 0.493 for the Web-Mining technique. The Wilcoxon-Signed rank test with continuity correction failed to show a significant improvement when comparing the classifier results to VSM ones (p-value = 0.05955).

An analysis of the results suggests that the web-mining approach works best when regulations describe specific practices or topics which are well understood and are clearly described in publicly retrievable documents. The Unique User Identification (UUI) regulation is an example of a HIPAA regulation which underperformed with the web-mining approach. Its MAP score dropped from 0.526 with VSM to 0.251 with web-mining approach. An analysis of the modified query showed that the web-mining approach retrieved a number of very general terms such as *mail*, *card*, and *com* from the online documents which caused a drop in precision. Integrity Controls (IC) is another interesting case. While the original VSM score was very low at 0.144, the web-mining approach failed to deliver much improvement because the original IC query was reduced to a single word *integ* in the modified query. This indicates that few commonalities were identified across the retrieved documents.

One of the primary benefits of web-mining is that no upfront training set is required. It can therefore be used without much prior human effort. However, our experiments indicate that it showed only limited, statistically insignificant, improvements. On the other hand it could be a useful option to add into a tracing tool, perhaps as a feature for recommending additional terms to include in a human modified trace query. The user could then accept or decline the recommendations for improving the trace query.

7 Ontology-Supported Traceability (OST)

Both the Classification and Web-Mining techniques add or replace terms in the original trace query without understanding the underlying semantics of those terms. For example, a new term such as *transaction* was added by the classifier to the Audit Control (AUD) query, thereby creating an effective bridge from the AUD regulation to the requirement stating that

"System will support transaction logs that will include the MID of the editor, transaction type and transaction date". This requirement was missed by the baseline VSM method.

A popular way to address term mismatch problems in general Q&A domains is through building and leveraging a domain ontology. An ontology provides formal definitions of concepts, identifies synonyms, expands acronyms, and captures relationships such as *is-a* and *part-of*. A *domain* ontology, as its name implies, captures the specific terminology and meanings that are common to a specific domain (Guizzardi 2010). For example, consider the requirement in Table 2, which states that *Automatic timeout can be specified by location*. A domain ontology could capture the relationship between *timeout* and *predetermined time of activity*, thereby allowing a trace link to be created between this requirement and the HIPAA Automatic Logoff (AL) requirement. Similarly, in our previous AUD example, the ontology could capture the concept that an *Audit log* records *event transactions*.

The benefits of using Ontology for generating trace links have previously been proposed, but little empirical work has been performed to evaluate whether it is actually effective. For example, Hayashi et al. (2010) and Assawamekin et al. (2010) proposed using ontology to connect concepts in source and target artifacts through relationships such as *is-a*, *composed-of*, or *is equivalent to*, allowing trace links to be established based upon general and specific instances of concepts occurring across pairs of artifacts. Instead of permanently replacing the original query with a new one, a domain ontology is used to dynamically augment terms and phrases in the original query at query execution time.

There are two primary challenges related to ontology use in traceability. First, there are few appropriate ontologies available. Most existing ones are not readily applicable to traceability tasks because they tend to represent everyday objects and do not include the technical engineering concepts needed to trace requirements and other kinds of entities in software-intensive systems. Second, the domain information within the ontology must be leveraged to identify related concepts so that correct and accurate trace links can be constructed.

7.1 Ontology Construction

Ontologies used for traceability purposes must be custom built to support very specific domains of use, a task which is difficult to fully automate and costly to perform. For example, in the HIPAA domain, there is a long-term community effort underway to build a technical ontology to support data queries (Grando and Schwab 2013; Grando et al. 2012). The partially constructed ontology defines entities such as *Subject* is a superclass of *Patient* and *Investigator*, *Consent*, *Operation*, and *HealtCareActivity*. Entities are connected through a fairly extensive set of relations that include *labRecord*, *canPerform-Operation*, *hasCreator*, and *shareWith*. For example, a *consent* hasSubject *Patient*. Further it *hasPolicyResults* of type *ConsentRule* which in turn include subclasses of *Permission* and *Obligation*.

Consider the example depicted in Fig. 7. In both requirements R1 and R2 a trace link should be created to the HIPAA automatic logoff regulation. In the case of R1, terms in the requirement such as "terminate" and "inactive" also occur in the HIPAA regulation, and a trace link can be established using a basic approach such as VSM. In the second case, there are insufficient shared terms; however, if an ontology contains the facts depicted for "time", "terminate", "computer", and "inactive" (among others), then the requirement can be matched to the regulatory code despite only sharing one non-trivial term (session). *Work Station* is associated with *Computer* via the is-a relationship in the ontology, *idle* is associated with *inactive*, and *minutes* is replaced with *time*. While numerous other associations are possible, these are the ones which bridge the gap between the HIPAA regulation and

HIPAA Automatic Logoff Regulation
Implement electronic procedures that terminate an electronic session after a predetermined time of inactivity.

R1: **Terminate** the **session** if the computer is **inactive** for 5 minutes.

R2: End the **session** if the work station is idle for 5 minutes.

R2': End the **session** if the ~~work station~~ [**computer**] is ~~idle~~ [**inactive**] for 5 ~~minutes~~ [**time**].

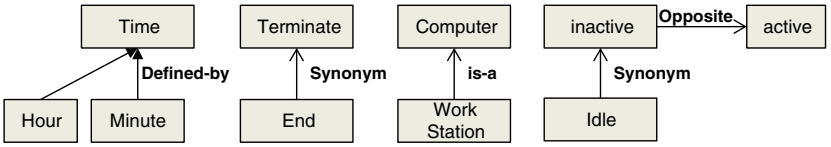


Fig. 7 Example showing how a small ontology bridges the gap between a regulation and a requirement

the requirement. The trace link is correctly established because of these concepts in the ontology.

In an ideal world, pre-constructed ontologies for a diverse set of technical domains would be readily available to support Software Engineering activities such as traceability. However, this is not the reality, and furthermore, building a sufficiently complete ontology for a specific domain is likely to produce conceptual facts which may be superfluous for tracing purposes. For purposes of the study reported in this paper we therefore manually created the ontology needed to support traceability in a *core project* and then used the resulting ontology for tracing purposes in the remaining projects. We recruited three DePaul Graduate students to perform the task of reviewing the HIPAA regulations and their trace links in the *core project* and identifying domain concepts that explain the link.

We provided a semi-automated tool which was designed to assist in this task. First, we used the Stanford Parser (Klein and Manning 2003), to extract nouns, verbs, and noun phrases from each HIPAA regulation and its linked requirement. We refer to these as the artifact's **Key Contents** KC. Next, for each pair of source and target artifacts related by a validated trace link from the core project, we generated candidate facts represented by pairs of KC_{Source} and KC_{Target} elements. For example, given a HIPAA regulation containing the phrases *audit log* and *record*, and a linked requirement containing terms *transaction* and *update*, we used our tool to explore the four candidate associations of *audit log - transaction*, *audit log - update*, *record-transaction*, and *record-update*. Finally, we applied a standard Association Rule Mining (ARM) algorithm (Agrwal and Srikant 1994) to compute the likelihood of each candidate fact, and then used this score to rank the candidate facts.

ARM was applied as follows. Given a set of connected artifact pairs T, a set of terms and noun phrases, i.e., items $I = \{I_1, I_2, \cdots, I_k\}$, and an item set $is \subseteq I$, let $T_{fi} \subseteq T$ be the set of transactions that have all the items in is. The *support* of the item set is is defined as $\sigma(is) = |T_{is}|/|T|$. Item sets that satisfy a predefined support threshold are referred to as *frequent item sets*. An association rule r is expressed in the form $X \implies Y(\sigma_r, \alpha_r)$, where $X \subseteq T$ and $Y \subseteq T$ are item sets, σ_r is the support of the item set $X \cup Y$, and α_r is the *confidence* for the rule r given by $\sigma(X \cup Y)/\sigma(X)$. The *lift* measure is defined as $\sigma(X \cup Y)/(\sigma(X) \times \sigma(Y))$.

Candidate facts were presented to the user ordered by *lift*. The user could accept or reject the candidate link. For all accepted CFs the user specifies relations as either *hierarchical*, *compositional*, and *equivalence* or could create a new type of relation. The user also assigned a *confidence rating* to each accepted fact. This is helpful because certain facts are likely to always hold, whereas others are only true sometimes. For example, the domain fact that 'access control' is parent of 'role-based access control' is generally true and could be ascribed a confidence rating of 1. On the other hand, the domain fact 'activity' has feature 'event type' is a less general relationship and is assigned a lower confidence score (in this case 0.5). In our current ontology, confidence ratings are determined by the user, utilizing three possible scores of 1, 0.8, or 0.5. As reported in the following section, our experimental results showed these subjective ratings to be quite effective.

Domain facts were saved in Protégé (Stanford 2013), which is a free, open-source ontology editor developed by Stanford University School of Medicine. The KC_{Source} and KC_{Target} for each fact are represented as entity names connected through the hierarchy of ontology classes or properties of those classes. Part of the domain ontology created from trace links between HIPAA and CCHIT (core project) is shown in Fig. 8. It is important to note that even with semi-automated support for ontology creation, the process is still currently human intensive and influenced by individual user decisions.

7.2 Leveraging Ontology to Generate Trace Links

Once ontology has been built for a domain, it can be used during the trace link generation process to expand terms and phrases. Our ontology-enhanced technique is built over the standard VSM model. Two KCs in the ontology are considered to be semantically related if they are connected either directly or indirectly. The *Semantic Relatedness* (SR) of two entities can be measured in several different ways. In this paper we consider two techniques which we define as *Information Content (IC)* based and *Directly Connected Domain Facts (DCF)*.

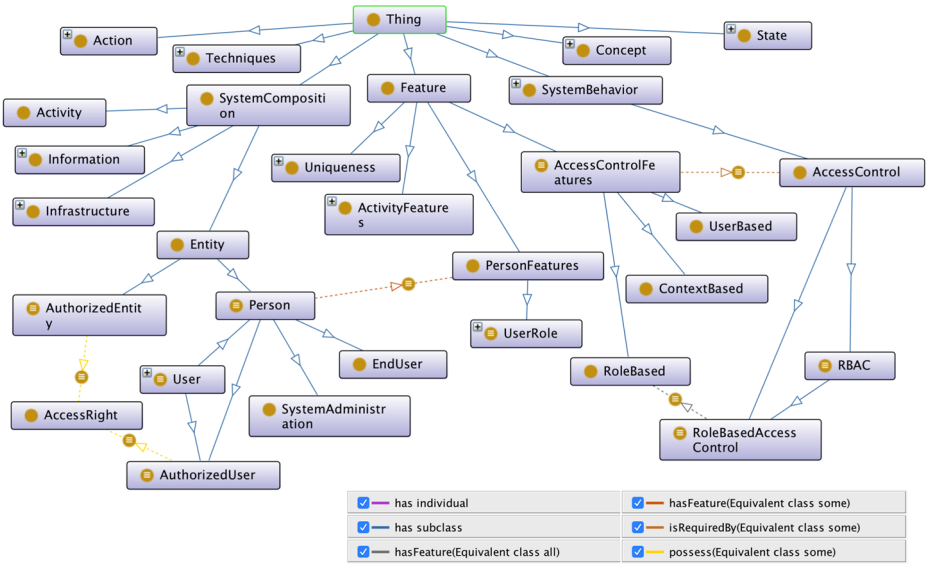


Fig. 8 Part of the ontology generated from trace links between HIPAA and CCHIT

Information Content (IC) based measures were proposed by (Lin 1998). They treat the ontology as a taxonomy and compute SR as follows:

$$SemanticRelatedness = 2 \times logP(E_0)/(logP(E_1) + logP(E_2)) \quad (9)$$

where E_0 is the Lowest Common Ancestor of E_1 and E_2 and $P(E_x)$ is the probability that a randomly selected entity belongs to E_x.

Directly Connected Domain Facts (DCF) consider only directly-related entities and weight their associations according to the user-generated confidence scores.

To process a query, we first check to see if it contains any domain entity names (e.g. "RoleBased" or "Infrastructure" as depicted in Fig. 8). If entity names exist in the query they are augmented with their associated ontology concepts. Furthermore, a weighting of $SR * w_{SA}$ is applied, where w_{SA} is an additional weight for controlling the influence of the concepts associated through the ontology. For experimental purposes we evaluated each of the semantic-relatedness algorithms at weights of 0.5 and 1.0. Following this step, both the query and documented are represented in the vector space alongside additional dimensions corresponding to their associated domain entities. During the trace retrieval phase, the similarity between the query and the document is then calculated as the cosine of the angle between query and document vectors weighted by the standard $tf - idf$ scheme. Examples of domain facts used during the tracing process are reported in Table 12.

Table 12 Using ontologies to match regulations and requirements at runtime

Reg	Sample REQ traced to reg	Used domain facts
AC	The credentials system employs a standards compliant **RBAC** model so each **user** has **permissions** appropriate to their role	Access Control is *parent of* RBAC; Person is *equivalent to* user; Access rights is *child of* permission
AL	Automatic **timeout** can be specified by role	Automatic logoff *has feature of* timeout
AUD	The system **audit trail** shall automatically capture system data changes, edits, charged, credits by all **users**	Audit *requires* audit trail; Activity *is performed by* user
EAP	When access to a chart is restricted and the **break the glass** has occurred, the system shall provide the ability to audit this override	Emergency *is a result of the action* break the glass
IC	The system shall provide the ability to retain data until otherwise purged, deleted, archived or otherwise deliberately **removed**	Dispose *is equivalent to* remove
PA	System will generate a secret key - patient's initial **password**	Authentication *is parent of* password
SED	The system, when storing PHI on any device intended to be portable/removable, shall support use of a standards based encrypted format using triple-DES, or the Advanced **Encryption Standard**, or their successors	Encryption *requires* encryption standard
TED	Everything is transmitted over SSL which offer encryption but also **hashes** to prevent modification in transit	Encrypt *has technique* hash
TS	The system shall ensure secure **electronic messaging** with patients	Electronic communication network *is required by* electronic messaging
UUI	System will generate a **unique** Medical User Identification number	Identify *requires* uniqueness; Track *requires* uniqueness

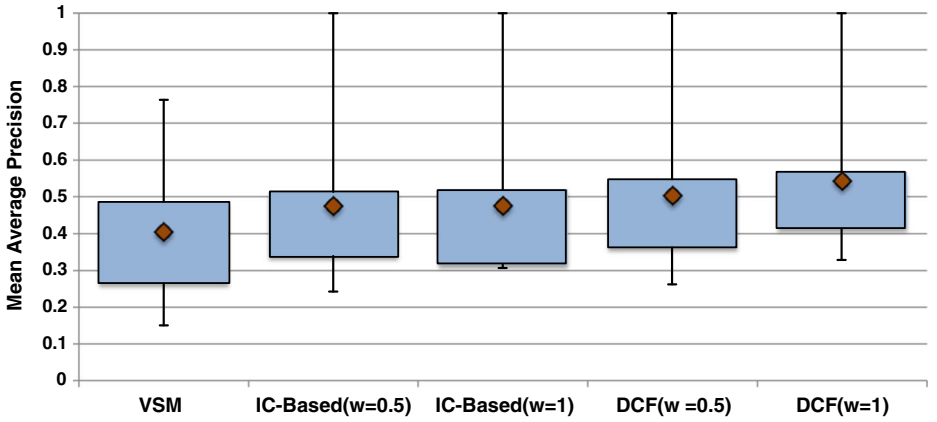


Fig. 9 A Comparison of VSM versus two ontology-based techniques with different weightings

7.3 Comparison of Two Ontology-Based Techniques and VSM

We evaluated the efficacy of the IC and DCF techniques on the overall quality of generated trace links. For experimental purposes, we constructed an ontology using the CCHIT dataset as the *core project* as this is the largest dataset and therefore most likely to generate a broad set of generally useful domain facts. The generated ontology was then used to support the generation of trace links between the HIPAA technical safeguards and each of the remaining datasets described in Table 1. Each of the Ontology-Based Techniques were evaluated and results are reported in terms of Mean Average Precision (MAP) in Fig. 9. Results indicate that the DCF approach for computing Semantic Relatedness outperformed the IC approach at both weightings, with its top performance observed when weighting was set to 1. For the winning method (i.e. DCF-Weighting 1.0) we report Recall, Precision, F2-Measure, and MAP scores against individual HIPAA technical safeguards in Table 13.

In Fig. 10 we compare the use of Ontology against our VSM baseline. As CCHIT was used to build the ontology, the ontology results are computed against the nine remaining

Table 13 Results from ontology-supported traceability method using the winning method of direct domain facts

HIPAA ID	Recall	Precision	F2	MAP
Access control (AC)	0.826	0.394	0.571	0.430
Automatic logoff (AL)	0.929	0.911	0.901	0.870
Audit trail (AUD)	0.932	0.648	0.774	0.666
Emergency access procedure (EAP)	1.000	1.000	1.000	1.000
Integrity controls (IC)	0.697	0.147	0.350	0.151
Personal authentication (PA)	0.966	0.328	0.577	0.393
Storage encryption & decryption (SED)	1.000	0.917	0.977	0.896
Transmission encryption & decryption (TED)	1.000	0.583	0.845	0.646
Transmission security (TS)	1.000	0.224	0.537	0.357
Unique user ID (UUI)	0.929	0.515	0.667	0.473

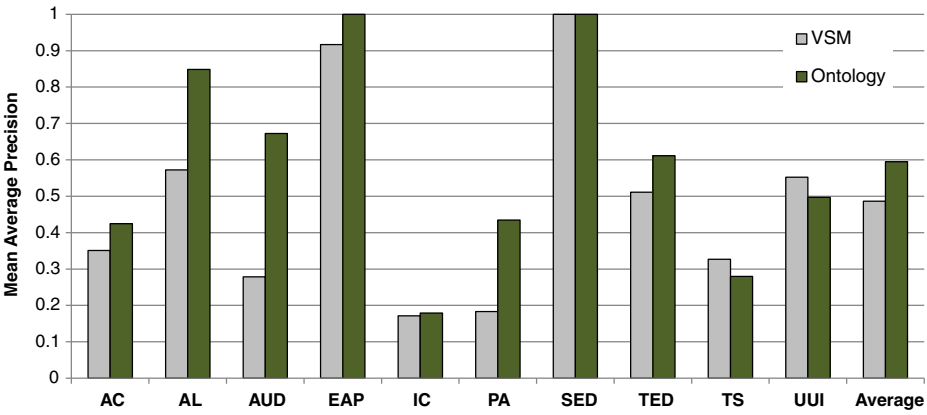


Fig. 10 A comparison of Ontology Use (using DCF-Weighting 1.0) versus the Baseline VSM approach. Note: The results shown here are for all projects except CCHIT. As CCHIT was used to build the ontology it is excluded from both Ontology and VSM results

projects only. For comparison purposes we therefore also report VSM results for only nine projects. Overall, we see that for seven out of the ten HIPAA regulations the use of the CHHIT-constructed ontology delivered improvements over VSM when applied to the remaining nine projects. At the level of the individual query, the ontology improved MAP scores for 36 queries, degraded it for 8 queries, and tied in 18 queries. We report results from our statistical analysis for the Ontology method in Section 8.

7.4 Impact of Ontology Source

The previous experiment used CCHIT, as the core project; however, to better understand the extent to which ontology built on one dataset could be applied across others from the same domain we conducted a second experiment, in which we systematically used each of the datasets as the core-project from which to construct an ontology. The resulting ontology was then used to generate trace links between HIPAA regulations and requirements in each of the remaining datasets. Results are reported in Table 14.

The CCHIT-constructed ontology outperformed the others in eight of the nine cases, the only exception being the Lauesen dataset which performed best on the WorldVista ontology. CCHIT's winning performance is not surprising seeing as it contains more artifacts and trace links than the other datasets, and therefore increases the chance of documenting domain facts used in other projects. However, while the CCHIT-built ontology outperforms the other ones, it is interesting to note that there were only six cases out of 90 in which use of any ontology delivered worse results than the use of VSM alone. One example is the case of applying the PracticeOne ontology to ClearHealth, where the MAP score dropped from 0.352 for VSM to 0.346.

8 Comparing Machine Learning, Web-Mining, and Ontology-Supported Approaches

In this paper we have presented three distinct techniques for augmenting queries, namely (1) a machine learning approach in which a classifier was trained to recognize requirements

Table 14 MAP scores achieved when ontology was constructed using a source dataset and applied against the target dataset

		Source dataset (used for constructing ontology)									
Target dataset	VSM	Care2x	CCHIT	ClearHealth	Physician	iTrust	TrialsImpls	PatientOS	PracticeOne	Lauesen	WorldVistA
Care2x	0.764	N/A	**1.000**	0.875	0.792	1.000	0.889	0.792	0.889	0.889	0.806
CCHIT	0.331	0.359	N/A	0.369	0.346	0.359	0.348	0.375	0.362	0.373	**0.380**
ClearHealth	0.352	0.511	**0.541**	N/A	0.378	0.370	0.375	0.484	0.346	0.289	0.438
Physician	0.265	0.265	**0.328**	0.271	N/A	0.280	0.320	0.274	0.265	0.288	0.288
iTrust	0.486	0.535	**0.593**	0.484	0.516	N/A	0.551	0.507	0.526	0.523	0.569
TrialImpls	0.411	0.431	**0.515**	0.449	0.425	0.455	N/A	0.443	0.422	0.419	0.433
PatientOS	0.560	0.508	**0.568**	0.544	0.560	0.498	0.565	N/A	0.498	0.541	0.543
PracticeOne	0.150	0.183	**0.411**	0.216	0.150	0.225	0.161	0.145	N/A	0.300	0.150
Lauesen	0.395	0.495	0.515	0.361	0.394	0.421	0.424	0.395	0.426	N/A	**0.523**
WorldVistA	0.260	0.343	**0.413**	0.208	0.243	0.264	0.260	0.310	0.260	0.283	N/A
# Average		0.403	0.542	0.419	0.422	0.43	0.432	0.413	0.443	0.433	0.458
# Domain Facts		6	50	13	10	10	17	8	7	14	10

Bold entries emphasize the winning data source for creating ontology

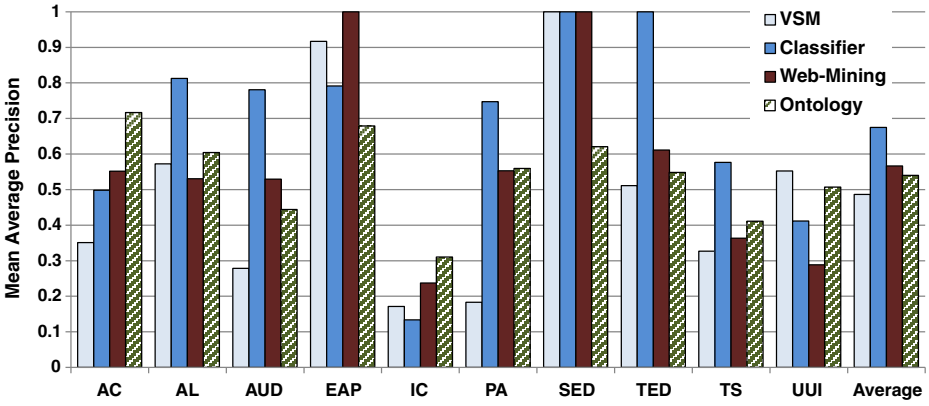


Fig. 11 Comparison of three techniques by HIPAA type

related to specific HIPAA regulations, (2) a web-mining approach in which queries were augmented with knowledge retrieved from mined domain documents, and finally (3) the ontology-supported approach. We now make a head-to-head comparison of their effectiveness for the HIPAA datasets.

8.1 Pairwise Statistical Analysis

Because our ontology-based approach requires one dataset (namely CCHIT) to be set aside, we perform this direct comparison using only nine datasets and exclude CCHIT from reported results for all four techniques. We report MAP results categorized by both HIPAA regulations in Fig. 11 and projects in Fig. 12.

Examining the results by project, we can see that the classifier was the overall winner in four projects (Consultations, iTrust, PatientOS, and Practice One), Web-Mining won in only one project (Clear Health), ontology won in one project (TrialImp), and the classifier

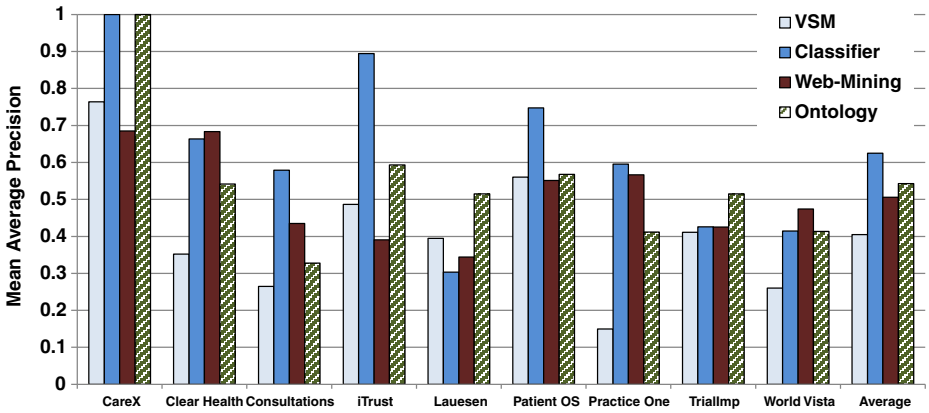


Fig. 12 Comparison of three techniques by Project

tied with ontology in one project (CareX). Examining results by HIPAA type, we observe that the classifier won in five cases (AL, AUD, PA, TED, and TS), Web-Mining won in one case (EAP), and ontology won in one case (AC). Three techniques of VSM, Classifier, and Web-mining tied in one case (SED).

Because we wished to compare all four techniques against one another we conducted the following statistical tests. First we performed the Kruskal-Wallis test to determine if statistically significant differences exist between our four techniques. The test achieved chi-squared = 9.1426, df = 3, p-value = 0.02745 which indicates that distinct results were achieved. We then performed a pairwise comparison using Pairwise Wilcoxon test with Bonferroni *p*-value adjustment. The Bonferroni adjustment reduces the overall likelihood of incorrectly rejecting one or more null hypotheses by chance due to making multiple comparisons. Results, which are depicted in Table 15 indicate the only statistical difference is observed between the VSM method and Machine Learning method.

8.2 Discussion of Results

Our reported results clearly show that in the case of the HIPAA datasets presented in this paper, the classifier performed best followed by the ontology-based approach. However, the HIPAA regulations exhibit very specific characteristics,which will not hold across all regulations. HIPAA regulations are relatively simple, for the most part describe distinct concepts, and are applied across multiple projects making it plausible to construct a training set which can be used by the classifier. Other regulations, such as Sarbanes-Oxley (Sarbanes-Oxley 2002), the Payment Card Industry Data Security Standard (PCI DSS) (PCI 2006), and other types of healthcare related regulations such as ISO/TS 21547:2010, Health informatics security requirements for archiving of electronic health records (ISO 2010), have quite similar characteristics, and therefore may be likely to perform in a similar way.

It is clear that characteristics of different types of regulations could strongly influence the choice and viability of trace query augmentation solutions. While there are many regulations that are similar in nature to HIPAA ones, there are also others can contain more technical definitions, use 'legalese', and describe quite complex behaviours that include constraints and cross-references (Breaux and Rao 2013). Consider one example of a software-intensive Positive Train Control system operating in Canada which needs to comply to NORAK regulations. One NORAK regulation states that "Gate Arm Clearance Time from a stop equals the time for the design vehicle to accelerate and travel completely through the gate clearance distance" while another states that "The total distance the vehicle must travel to pass completely through the clearance distance ... is computed using the following formula: s = cd + L where: s = distance the road vehicle must travel to pass through the grade crossing clearance distance, (m) cd = grade crossing clearance distance L = length of the grade crossing design vehicle." Such regulations are clearly different in nature to the HIPAA technical safeguards, and even if training sets were available, the nuanced differences between

Table 15 Pairwise Wilcoxon test result

	VSM	MachineLearning	Ontology-Supported
MachineLearning	0.035	–	–
Ontology-supported	0.272	1.000	–
Webmining	0.730	0.553	1.000

such regulations would make it relatively difficult to train an effective classifier even though ontology use may still be effective (Guo et al. 2013, 2014).

We document tradeoffs, startup costs, applicability, and constraints for the three query augmentation techniques in Table 16.

9 Threats to Validity

In this section we describe specific threats to validity for the reported study. Such threats fall under the categories of construct validity, internal validity, and external validity (Wohlin et al. 2000).

9.1 Construct Validity

The goal of this study was to evaluate whether the augmented trace queries were more accurate than those generated using the original HIPAA regulations as the queries. To determine this we first needed to establish an accurate reference set that contained the correct set of trace links from HIPAA regulations to each of the ten software requirements specifications. Such reference sets were not provided to us, and we therefore had to construct them ourselves. The data sets were initially collected and constructed by a Graduate Information Systems student and trace links were collaboratively and systematically created and vetted by four members of our research team. Unfortunately creating trace links is not an exact science, and while some trace links are obvious, there are other cases in which judgment calls must be made. These borderline cases were resolved through discussion. We point out that the presence of borderline cases is usually not caused by lack of expertise, as domain experts often disagree over these borderline cases too. To evaluate the quality of the generated trace links, we needed to measure the extent to which trace links matched the reference set. Although there are numerous metrics that can be used, we adopted the Mean Average Precision (MAP) metric described in formula (4). This is a well-accepted metric for evaluating trace results, and measures the extent to which the correct links were placed at the top of an ordered set of candidate links (Shin et al. 2015).

9.2 Internal Validity

The results reported from our experiments show that under certain circumstances one query performs better than another one. The primary focus of the work in this paper was on reporting the methods and results of using three different enhancement techniques. We applied all three techniques to the same datasets and evaluated results using the same metrics. However, for each individual technique we made configuration decisions which may have influenced the results. In all cases, these decisions were made following rather extensive informal experimentation. We have documented these for each of the three techniques.

9.3 External Validity

External validity refers to the extent to which results from the study can be generalized across the entire domain of study. As discussed in the previous section we are not able to claim generalizability for tracing the diverse set of regulatory codes describing elements such as electrical, water, or power regulations, that apply across the systems in engineering domain.

Table 16 A summary of the costs, benefits, and constraints of applying various query augmentation techniques

Technique	Usage steps	Cost	Applicability	Constraints
Classification	Constructing a training set. Requires a tool (e.g. Weka or TraceLab) for training and utilizing a classifier	Training set construction took 125 person hours). Costs can be recouped if an organization builds multiple software products in a domain	Training set size must be sufficiently large, and requirements within the category must be relatively homogenous	ROI only realized if similar trace queries are applied across multiple projects
Web mining	Instrumenting the environment to search for relevant documents, analyze the text, and to identify replacement terms and phrases for the original query	No additional manual effort is required beyond initial instrumentation. No training set required	Works best when the topic(s) covered by the trace query are clearly defined and explained in publicly available web documents or pages	Unlikely to be effective for highly complex and/or relatively obscure queries (e.g. for Positive Train Control)
Ontology	Acquire or construct relevant domain ontology. Instrument the traceability environment to leverage ontology facts	Ontology construction for 11 HIPAA regulations using the CHITT dataset took 14 person hours (10 h identifying domain facts, and 4 h constructing ontology)	Broadly applicable wherever ontology is available	Domain-wide ontology construction is costly and effort intensive

The experiments and techniques described in this paper focused around tracing HIPAA security safeguards; however it is well known from the data mining literature that results are highly sensitive to subtle nuances of the datasets. The question is therefore to what extent the study of HIPAA regulations is generalizable across a broader set of regulatory codes such as Sarbanes-Oxley (Sarbanes-Oxley 2002), the Payment Card Industry Data Security Standard (PCI DSS) (PCI 2006), and other types of healthcare related regulations such as ISO/TS 21547:2010, Health informatics security requirements for archiving of electronic health records (ISO 2010). Due to the significant time and effort needed to construct datasets, traceability reference sets, and to build ontologies, we were only able to conduct experiments with HIPAA regulations. It is likely that different techniques may perform better on different types of regulatory codes. The contribution of this paper is therefore in presenting different query augmentation techniques and evaluating them within one specific context. Additional work is needed to evaluation them against a more diverse set of regulations.

On the other hand we claim our approach to be generalizable for tracing HIPAA regulations across a wide range of software requirements specifications. The reported experiments were based on 10 independent requirements specifications taken from entirely different sources. In all cases, the requirements text was extracted directly from source documents, and only very common words were added during the specification process that would either be ignored as stopwords or assigned insignificant weightings by the tf-idf algorithm. Furthermore, we have reported results for each individual project, and have shown that the benefits extend across most of the projects. The results also show that there are cases, i.e. specific projects, that do not trace as well as others, because they do in fact use terminology that is different from the HIPAA regulations and from the terminology used in other projects. This should not be seen as a threat to validity, but rather as a realistic observation and very real limitation of the effectiveness of any trace-retrieval technique.

10 Related Work

In this section we summarize key related work in the area of query augmentation– first describing baseline techniques, then presenting traceability solutions which leverage external knowledge to improve accuracy, and finally discussing the use of Ontology for tracing purposes.

10.1 Baseline Tracing Techniques

Many different researchers have investigated automated approaches for dynamically generating trace links using standard techniques such as the Vector Space Model (VSM) (Hayes et al. 2004; Hayes et al. 2006), Latent Semantic Indexing (Antoniol et al. 2002; Marcus and Maletic 2000; 2003), and Latent Dirichlet Allocation (LDA). Of these techniques there is no clear "winner" across a diverse collection of datasets (Lohar et al. 2013; Falessi et al. 2013; Lucia et al. 2012). Numerous enhancements have been suggested, but fully automated solutions rarely outperform the original methods. For example, Sultanov and Hayes (2013) proposed the use of reinforcement learning but were not able to achieve more than 55 % recall in either of the tested datasets. They also proposed the use of AI swarm technologies (Sultanov et al. 2011), and reported improvements over a VSM baseline when applied to one dataset but decreased accuracy in another. In our own prior work we proposed a variety of clustering techniques to enhance tracing results, but only observed small improvements in trace quality (Cleland-Huang et al. 2005).

10.2 Augmented Tracing Techniques

Another class of tracing techniques include additional information and/or guidance into the tracing process. These techniques include rule-based approaches (Spanoudakis et al. 2004), scenario and test case-based methods (Egyed 2003), event-based approaches (Cleland-Huang et al. 2003), and policy-based methods (Murta et al. 2006). However much of this work has focused on generating traces between documentation and code (Antoniol et al. 2000; Bennett and Rajlich 2000) or across artifacts such as requirements, design, code, and test cases within a project. Furthermore, techniques such as scenario or policy-based approaches are more appropriate for tracing within a project than for tracing external documents such as regulatory codes.

While the idea of using information retrieval methods to augment web-based search queries is not new, prior work has focused on user-defined queries for purposes of retrieving information. Such queries typically contain between 2.4 and 2.7 terms (Gabrilovich et al. 2009) and are repetitious across multiple users. In contrast, trace queries often contain 10-100 terms, utilize technical and legal terms, and are used infrequently by a small group of trace users. Despite these differences there is a large foundation of work in the area of query expansion that is relevant to the work we propose in this paper. For example, Hu et al., used Wikipedia to augment users' web-based queries (Hu et al. 2009); however based on our initial analysis of the traceability problem, websites such as Wikipedia, do not currently provide sufficient information to augment a rich and diverse set of domain specific traceability queries. Broder et al. (2007) and Shen et al. (2006) have used more general web knowledge to augment queries; however their techniques are primarily designed to expand a short query with additional and potentially disambiguating terms. In contrast, our approach needs to both add and remove terms from an original query.

Researchers in the area of Feature Location have also explored the use of query augmentation techniques. The problem they address is somewhat different from ours, because the target of their search is source code, which is highly structured. Feature location represents a subtly different task from requirements traceability. Its goal is to find source code *related to* a query, whereas the goal of tracing regulations to requirements is that of find requirements which contribute to the *satisfaction* of the regulations. Nevertheless, there are many commonalities, for example Hill et al. (2008) focused their feature location searches on noun phrases, verb phrases, and prepositional phrases extracted from method and variable names and organized into a hierarchy. User queries are expanded using the hierarchy. Abebe and Tonella (2015) extracted domain facts from the structure of the source code and stored them in a basic ontology. They allowed developers to use the ontology to formulate more precise queries. Finally, Dasgupta et al. (2013) used external documentation to augment the corpora of a search space in order to improve its searchability for traceability purposes. They demonstrated an increase in accuracy ranging from 8 to 31 %.

Other researchers such as Yurelki et al., proposed collaborative recommender systems which suggest query refinements to the user based on other users' similar queries (Yurekli et al. 2009). Others have augmented queries with different types of knowledge such as folksonomy or taxonomy tags (Nauman and Khan 2007). For example, Gabrilovich et al. (2009) classify initial search results according to an extensive taxonomy and then improve search results based on the prior history of queries in the designated categories. Unfortunately, trace queries cannot directly benefit from such techniques as they rely on a large user base and a historical log of related queries. In the area of feature location.

Hayes et al. (2006) and Lucia et al. (2006) both evaluated the use of the well-known Rocchio technique (Baeza-Yates and Ribeiro-Neto 1999) to capture relevance feedback in order to improve trace quality. The Rocchio algorithm, utilizes relevance feedback captured from an initial set of traceability links to modify the underlying representation of the query. We previously introduced the Direct Query Modification (DQM) approach (Cleland-Huang et al. 2010; Gibiec et al. 2010; Shin and Cleland-Huang 2012), which allows a user to directly modify the trace query by filtering out unwanted terms and adding additional terms and synonyms.

More recently Gervasi and Zowhgi used affinity mining for traceability purposes. Their approach also uses a training set of trace links and uses these to learn affinities between terms in source and target artifacts. Future trace links are generated based on this set of learned affinities (Gervasi and Zowghi 2014). Their approach is very similar to a technique we previously presented which leverages relationships between terms that occurred in linked source and target artifacts to generate future trace links (Dietrich et al. 2013). Gervasi and Zowghi evaluated their work against only one dataset - CM1 which is also used in our dataset. They also used a leave-one-out experimental design and their approach returned precision values of 85 % at recall of 90 %. Given our analysis of the CM1 dataset and the dearth of association rules we were able to mine, we are unable to explain or duplicate such results.

10.3 Ontology in Traceability

There is a small body of work in the area of ontology use for Traceability. Kof et al. (2010) extracted domain-specific concepts from the set of traceable artifacts, and then used WordNet to find synonyms in order to map similar concepts and to establish trace links. The major contribution of their work is their approach for extracting an ontology from a requirements specification (Kof et al. 2010); however they did not provide a rigorous analysis to evaluate whether the use of the ontology actually improves the quality of the generated trace links. Li at al. also explored the use of an ontology for traceablity purposes (Li and Cleland-Huang 2009). Their approach used knowledge from the ontology to establish connections between concepts in source and target artifacts. The strength of the dependency was related to the distance between concepts in the ontology, and the overall similarity score between pairs of artifacts was increased accordingly. This approach was tested on only one dataset and led to only small improvements in trace accuracy.

Existing tracing tools such as Poirot (Lin et al. 2006) include capabilities to expand acronyms and utilize lists of matching words in an attempt to bridge the semantic gap. In most cases, these synonyms and acronyms are extracted directly from project documents. Such approaches are not designed to capture the rich set of relationships and inference rules that can be modeled in an expert system.

Zhang et al. published a paper entitled “Ontological approach for the semantic recovery of trace links between software artefacts” (Zhang et al. 2008); however ontology is used to represent concepts such as *design patterns*, *sentences*, *paragraphs*, and *classes*, as opposed to domain concepts such as *train* or *signal* found in the domain being traced.

Assawamekin et al. (2010) utilized Natural Language Processing techniques to construct separate ontologies from each stakeholder perspective and then to discover a mapping between them . The ontology is expressed in first-order logic and the mapping is solved by a SAT-solver. Their approach is designed to create links between different stakeholder perspectives; however the authors provided only a simple example, and did not empirically evaluate their approach.

11 Conclusion

This paper has described three techniques for modifying and/or augmenting trace queries in order to generate more accurate trace links. The first approach trained a classifier to recognize relevant requirements. This technique proved to be highly effective in cases where the regulations were sufficiently focused and associated requirements were relatively cohesive. As regulatory codes apply to hundreds, and even thousands of different software systems, it seems worthwhile expending the time and effort to train a trace classifier to perform the tracing task. The benefits could be returned many times over. Ideally such trace classifiers could be provided by regulatory bodies in conjunction with the regulatory codes as this could result in very significant savings for organizations needing to demonstrate compliance or otherwise certify their products.

The second approach presented in this paper used a web-mining technique to discover terms that could be used to replace an original trace query. This approach returned mixed results, and only showed improvements for a subset of the generated trace links. Given the inherently labor-intensive nature of traceability, we believe our web-mining technique could be usefully offered as an optional feature in tracing tools such as Poirot, to assist the user in learning a new set of query terms.

The third approach uses ontology to enhace concepts in the query. This approach holds signficant promise for future success in achieving the goal of accurate, automated traceability (Cleland-Huang and Guo 2014). Whereas, machine learning and web-mining approaches treat trace queries as a bag of words which they expand or modify into a different bag of words, ontology-based approaches aim to add semantics which enable higher levels of reasoning. Ontology construction is notoriously effort intensive; however, we have demonstrated the potential viability of leveraging trace links in one project to build ontology that can be reused across multiple projects—thereby recouping the costs involved in the initial ontology construction effort.

In conclusion, our work makes a novel contribution to the area of tracing requirements to regulatory codes. It highlights the term-mismatch problem and its impact upon traceability, and explores the benefits and tradeoffs of three different query augmentation techniques. Our ongoing work (Guo 2016) focuses on automating the generation of domain ontologies in order to further improve the accuracy of the tracing task.

Acknowledgments The work described in this paper was supported by US National Science Foundation grants CCF-1319680 and CCF-0447594.

References

Abebe SL, Tonella P (2015) Extraction of domain concepts from the source code. Sci Comput Program 98:680–706

Agrwal R, Srikant R (1994) Fast algorithms for mining association rules. In: Proceedings of the 20th International Conference on very large data Bases (VLDB'94). Santiago, Chile, pp 487–499

Antoniol G, Canfora G, de Lucia A, Casazza G (2000) Information retrieval models for recovering traceability links between code and documentation. In: ICSM '00: Proceedings of the international conference on software maintenance (ICSM'00). IEEE Computer Society, Washington, DC, USA, p 40

Antoniol G, Canfora G, Casazza G, De Lucia A, Merlo E (2002) Recovering traceability links between code and documentation. IEEE Trans Softw Eng 28(10):970–983

Assawamekin N, Sunetnanta T, Pluempitiwiriyawej C (2010) Ontology-based multiperspective requirements traceability framework. Knowl Inf Syst 25(3):493–522

Baeza-Yates RA, Ribeiro-Neto BA (1999) Modern information retrieval. ACM Press/Addison-Wesley

Bennett KH, Rajlich VT (2000) Software maintenance and evolution: a roadmap. In: ICSE '00: Proceedings of the Conference on the future of software engineering. ACM, New York, NY, USA, pp 73–87

Berenbach B, Gruseman D, Cleland-Huang J (2010) Application of just in time tracing to regulatory codes. In: Proceedings of the conference on systems engineering research. Stevens Institute of Technology. Holbroken, NJ

Breaux TD, Rao A (2013) Formal analysis of privacy requirements specifications for multi-tier applications. In: 21st IEEE international requirements engineering conference, RE, 2013, Rio de Janeiro-RJ, Brazil, July 15–19, 2013, pp 14–20

Broder AZ, Fontoura M, Gabrilovich E, Joshi A, Josifovski V, Zhang T (2007) Robust classification of rare queries using web knowledge. In: SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval. ACM, New York, NY, USA, pp 231–238

Cleland-Huang J, Guo J (2014) Towards more intelligent trace retrieval algorithms. In: (RAISE) Workshop on realizing artificial intelligence synergies in software engineering

Cleland-Huang J, Chang CK, Christensen M (2003) Event-based traceability for managing evolutionary change. IEEE Trans Softw Eng 29(9):796–810

Cleland-Huang J, Settimi R, Duan C, Zou X (2005) Utilizing supporting evidence to improve dynamic requirements traceability. In: 13th IEEE international conference on requirements engineering (RE 2005), 29 August–2 September 2005, Paris, France, pp 135–144

Cleland-Huang J, Settimi R, Zou X, Solc P (2006) The detection and classification of non-functional requirements with application to early aspects. In: RE, pp 36–45

Cleland-Huang J, Berenbach B, Clark S, Settimi R, Romanova E (2007) Best practices for automated traceability. Computer 40(6):27–35

Cleland-Huang J, Settimi R, Zou X, Solc P (2007) Automated detection and classification of non-functional requirements. Requir Eng 12(2):103–120

Cleland-Huang J, Czauderna A, Gibiec M, Emenecker J (2010) A machine learning approach for tracing regulatory codes to product specific requirements. In: ICSE '10: Proceedings of the 32nd ACM/IEEE international conference on software engineering. ACM, New York, NY, USA, pp 155–164

CoEST (2008) Center of excellence for software traceability, http://www.traceabilitycenter.org

Cuddeback D, Dekhtyar A, Hayes JH (2010) Automated requirements traceability: the study of human analysts. In: RE'10: Proceedings of the IEEE international requirements engineering conference. IEEE

Dasgupta T, Grechanik M, Moritz E, Dit B, Poshyvanyk D (2013) Enhancing software traceability by automatically expanding corpora with relevant documentation. In: 2013 IEEE international conference on software maintenance, Eindhoven, The Netherlands, September 22–28, 2013, pp 320–329

Dietrich T, Cleland-Huang J, Shin Y (2013) Learning effective query transformations for enhanced requirements trace retrieval. In: 2013 28th IEEE/ACM International Conference on Automated Software Engineering, ASE, 2013, Silicon valley, CA, USA. November 11–15, 2013, pp 586–591

Egyed A (2003) A scenario-driven approach to trace dependency analysis. IEEE Trans Softw Eng 29(2):116–132

FAA. AC20-115C. Do-178c: Software considerations in airborne systems and equipment certification

Falessi D, Cantone G, Canfora G (2013) Empirical principles and an industrial case study in retrieving equivalent requirements via natural language processing techniques. IEEE Trans Software Eng 39(1):18–44

Gabrilovich E, Broder A, Fontoura M, Joshi A, Josifovski V, Riedel L, Zhang T (2009) Classifying search queries using the web as a source of knowledge. ACM Trans Web 3(2):1–28

Gervasi V, Zowghi D (2014) Supporting traceability through affinity mining. In 2014 IEEE 22nd International Requirements Engineering Conference (RE), Karlskrona, pp 143–152

Gibiec M, Czauderna A, Cleland-Huang J (2010) Towards mining replacement queries for hard-to-retrieve traces. In: ASE '10: Proceedings of the IEEE/acm international conference on automated software engineering. ACM, New York, NY, USA, pp 245–254

Gotel OCZ, Finkelstein ACW (1994) An analysis of the requirements traceability problem. pp 94–101

Gotel O, Finkelstein A (1997) Extended requirements traceability: results of an industrial case study. In: RE '97: Proceedings of the 3rd IEEE international symposium on requirements engineering. IEEE Computer Society, Washington, DC, USA, p 169

Grando A, Schwab R (2013) Building and evaluating an ontology-based tool for reasoning about consent permission. In: AMIA annual symposium proceedings, pp 514–523

Grando M, Boxwala A, Schwab R, Alipanah N (2012) Ontological approach for the management of informed consent permissions. In: 2012 IEEE second international conference on healthcare informatics imaging and systems biology (HISB), pp 51–60

Gruber TR (1993) A translation approach to portable ontology specifications. Knowledge acquisition 5(2):199–220

Guizzardi G (2010) Theoretical foundations and engineering tools for building ontologies as reference conceptual models. Semantic Web 1(1–2):3–10

Guo J (2016) Ontology learning and its application in software-intensive projects. In: Proceedings of the 38th international conference on software engineering, ICSE 2016, Austin, TX, USA, May 14–22, 2016 - Companion Volume, pp 843–846

Guo J, Cleland-Huang J, Berenbach B (2013) Foundations for an expert system in domainspecific traceability In 2013 21st IEEE International Requirements Engineering Conference (RE), Rio de Janeiro 2013, pp 42-51

Guo J, Monaikul N, Plepel C, Cleland-Huang J (2014) Towards an intelligent domain-specific traceability solution. In: ACM/IEEE international conference on automated software engineering, ASE '14, Vasteras, Sweden - September 15–19, 2014, pp 755–766

Hayashi S, Yoshikawa T, Saeki M (2010) Sentence-to-code traceability recovery with domain ontologies. In: Han J, Thu TD (eds) APSEC, pp 385–394. IEEE Computer Society

Hayes JH, Dekhtyar A, Sundaram SK (2006) Advancing candidate link generation for requirements tracing: the study of methods. IEEE Trans Softw Eng 32(1):4–19

Hayes JH, Dekhtyar A, Sundaram SK, Howard S (2004) Helping analysts trace requirements: an objective look. In: RE '04: Proceedings of the requirements engineering conference, 12th IEEE international. IEEE Computer Society, Washington, DC, USA, pp 249–259

Hill E, Fry ZP, Boyd H, Sridhara G, Novikova Y, Pollock LL, Vijay-Shanker K (2008) AMAP: Automatically mining abbreviation expansions in programs to enhance software maintenance tools. In: Proceedings of the 2008 international working conference on mining software repositories, MSR 2008 (Co-located with ICSE), Leipzig, Germany, May 10–11, 2008, Proceedings, pp 79–88

Hu J, Wang G, Lochovsky F, Sun J-T, Chen Z (2009) Understanding user's query intent with wikipedia. In: WWW '09: Proceedings of the 18th international conference on world wide web. ACM, New York, NY, USA, pp 471–480

ISO (2010) Iso/ts 21547:2010, health informatics security requirements for archiving of electronic health records. International Organization for Standards TC 215 Health Informatics, http://www.iso.org/iso/home.htm (Last accessed 12/20/10)

Klein D, Manning CD (2003) Accurate unlexicalized parsing. In: Proceedings of the 41st annual meeting on association for computational linguistics - vol 1. ACL '03. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 423–430

Kof L, Gacitua R, Rouncefield M, Sawyer P (2010) Concept mapping as a means of requirements tracing. In: MaRK'10)

Li Y, Cleland-Huang J (2009) Ontology-based trace retrieval. In: Traceability in emerging forms of software engineering (TEFSE2013. San Francisco, USA

Lin D (1998) An information-theoretic definition of similarity. In: ICML., vol 98, pp 296–304

Lin J, Lin CC, Cleland-Huang J, Settimi R, Amaya J, Bedford G, Berenbach B, Khadra OB, Duan C, Zou X (2006) Poirot: a distributed tool supporting enterprise-wide automated traceability. In: RE, pp 356–357

Lohar S, Amornborvornwong S, Zisman A, Cleland-Huang J (2013) Improving trace accuracy through data-driven configuration and composition of tracing features. In: Joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on the foundations of software engineering, ESEC/FSE'13, Saint Petersburg, Russian Federation, August 18–26, 2013, pp 378–388

Lucia AD, Oliveto R, Sgueglia P (2006) Incremental approach and user feedbacks: a silver bullet for traceability recovery. IEEE International Conference on Software Maintenance 0:299–309

Lucia AD, Marcus A, Oliveto R, Poshyvanyk D (2012) Information retrieval methods for automated traceability recovery. In: Software and systems traceability., pp 71–98

Mäder P., Jones PL, Zhang Y, Cleland-Huang J (2013) Strategic traceability for safety-critical projects. IEEE Softw 30(3):58–66

Mahmoud A, Niu N (2015) On the role of semantics in automated requirements tracing. Requir Eng 20(3):281–300

Marcus A, Maletic JI (2000) Using latent semantic analysis to identify similarities in source code to support program understanding. In: ICTAI '00: Proceedings of the 12th IEEE international conference on tools with artificial intelligence. IEEE Computer Society, Washington, DC, USA, p 46

Marcus A, Maletic JI (2003) Recovering documentation-to-source-code traceability links using latent semantic indexing. In: ICSE '03: Proceedings of the 25th international conference on software engineering. IEEE Computer Society, Washington, DC, USA, pp 125–135

Mirakhorli M, Cleland-Huang J (2016) Detecting, tracing, and monitoring architectural tactics in code. IEEE Trans Software Eng 42(3):205–220

Mirakhorli M, Shin Y, Cleland-Huang J, Çinar M (2012) A tactic-centric approach for automating traceability of quality concerns

Murta LGP, van der Hoek A, Werner CML (2006) Archtrace: Policy-based support for managing evolving architecture-to-implementation traceability links. In: ASE '06: Proceedings of the 21st IEEE/ACM International Conference on Automated Software Engineering. IEEE Computer Society, Washington, DC, USA, pp 135–144

Nauman M, Khan S (2007) Using personalizedweb search for enhancing common sense and folksonomy based intelligent search systems. In: WI '07: Proceedings of the IEEE/WIC/ACM international conference on web intelligence. IEEE Computer Society, Washington, DC, USA, pp 423–426

PCI (2006) Pci security standard quick reference guide. Payment Card Industry Security Guidelines https://www.pcisecuritystandards.org (Last accessed 12/30/10)

Porter M (1980) Porter's stemming algorithm, pp 130–137

Ramesh B, Jarke M (2001) Toward reference models for requirements traceability. IEEE Trans Softw Eng 27(1):58–93

Rempel P, Mäder P., Kuschke T, Cleland-Huang J (2015) Traceability gap analysis for assessing the conformance of software traceability to relevant guidelines. In: Software Engineering & Management 2015, Multikonferenz der GI-fachbereiche Softwaretechnik (SWT) und Wirtschaftsinformatik (WI), FA WI-MAW, 17. März - 20. März 2015, Dresden, Germany, pp 120–121

Salton G (1989) Automatic text processing: the transformation, analysis, and retrieval of information by computer. Addison-Wesley Longman Publishing Co., Inc., Boston

Salton G, McGill M (1986) Introduction to modern information retrieval. McGraw-Hill, New York

Salton G, Wong A, Yang CS (1975) A vector space model for automatic indexing. Commun ACM 18(11):613–620

Sarbanes-Oxley (2002) A guide to the sarbanes-oxley act. The Sarbanes-Oxley Compliance Guide 2002, http://www.soxtoolkit.com/ (Last accessed on 12/30/10)

Shen D, Sun J-T, Yang Q, Chen Z (2006) Building bridges for web query classification. In: SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval. ACM, New York, NY, USA, pp 131–138

Shin Y, Cleland-Huang J (2012) A comparative evaluation of two user feedback techniques for requirements trace retrieval. In: SAC, pp 1069–1074

Shin Y, Hayes JH, Cleland-Huang J (2015) Guidelines for benchmarking automated software traceability techniques. In: 8th IEEE/ACM international symposium on software and systems traceability, SST, 2015, Florence, Italy, May 17, 2015, pp 61–67

Spanoudakis G, Zisman A, Pérez-Miñana E, Krause P (2004) Rule-based generation of requirements traceability relations. J Syst Softw 72(2):105–127

Stanford (2013) Protégé: Open source ontology editor

Sultanov H, Hayes JH (2013) Application of reinforcement learning to requirements engineering: requirements tracing. In: 21st IEEE International Requirements Engineering Conference, RE 2013, Rio de janeiro-RJ, Brazil, July 15–19, 2013, pp 52–61

Sultanov H, Hayes JH, Kong W (2011) Application of swarm techniques to requirements tracing. Requir Eng 16(3):209–226

Tufis D, Mason O (1998) Tagging romanian texts: a case study for qtag, a language independent probabilistic tagger. In: Proceedings of the first international conference on language resources and evaluation (LREC), pp 589–596

U.S. Food and Drug Administration (2002) General principles of software validation. U.S. Dept. of Health and Human Services 1:1

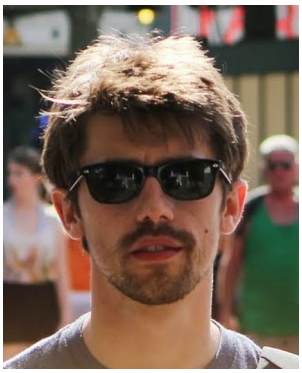
Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A (2000) Experimentation in software engineering: an introduction. Kluwer Academic Publishers, Norwell

Yurekli B, Capan G, Yilmazel B, Yilmazel O (2009) Guided navigation using query log mining through query expansion. In: NSS '09: Proceedings of the 2009 third international conference on network and system security. IEEE Computer Society, Washington, DC, USA, pp 560–564

Zhang Y, Witte R, Rilling J, Haarslev V (2008) Ontological approach for the semantic recovery of traceability links between software artefacts. Software, IET 2(3):185–203

Zou X, Settimi R, Cleland-Huang J (2010) Improving automated requirements trace retrieval: a study of term-based enhancement methods. Empirical Softw Engg 15(2):119–146

Jin Guo is a PhD candidate at the University of Notre Dame supervised by Dr. Jane Cleland-Huang. Her current research focuses on developing semantic enhanced software traceability solutions and mining domain knowledge from traceability data. Her interests include software traceability, requirement compliance, and application of machine learning and natural language processing techniques to support software engineering tasks.



Marek Gibiec graduated from DePaul University with his Masters in Computer Science. He currently works as an IT Professional at EYAccounting in Chicago.

Jane Cleland-Huang is a Professor in Software Engineering at the University of Notre Dame. Her research interests focus upon Software and Systems Traceability for Safety- Critical Systems with a particular focus on the application of machine learning and natural language processing techniques to solve large-scale software and requirements engineering problems.