# Extraction of Natural Language Requirements from Breach Reports Using Event Inference

Hui Guo*, Özgür Kafalı†, and Munindar P. Singh*
*Department of Computer Science, North Carolina State University, Raleigh, NC, USA
†School of Computing, University of Kent, Canterbury, Kent, UK
Email: hguo5@ncsu.edu, R.O.Kafali@kent.ac.uk, m.singh@ieee.org

***Abstract*—We address the problem of extracting useful information contained in security and privacy breach reports. A breach report tells a short story describing how a breach happened and the follow-up remedial actions taken by the responsible parties. By predicting sentences that may follow a breach description using natural language processing, our goal is to suggest security and privacy requirements for practitioners and end users that can be used to prevent and recover from such breaches. We prepare a curated dataset of structured short breach stories using unstructured breach reports published by the U.S. Department of Health and Human Services. We propose a prediction model for inferring held-out sentences based on Paragraph Vector, a document embedding method, and Long Short-Term Memory networks. The predicted sentences can suggest natural language requirements. We evaluate our model on the curated dataset as well as the ROCStories corpus, a collection of five-sentence commonsense stories, and find that the presented model performs significantly better than the baseline of using average word vectors.**

***Index Terms*—Event inference, Story Cloze Test, security and privacy requirements, breach reports, recurrent neural networks, Long Short-Term Memory architecture**

## I. Introduction

Natural language artifacts regarding software security and privacy contain useful information that requirements engineers can make use of, some of which are story-like reports that contain causal and logical structures of narrative events. For example, in the healthcare domain, the U.S. Department of Health and Human Services (HHS) is legally required to maintain and publish a dataset of breach reports that describe incidents in which protected health information (PHI) was missing, stolen, improperly disposed of, or impermissibly accessed or disclosed [1]. In addition, they record the actions that responsible parties have taken to prevent, detect, and recover from future breaches. The knowledge that resides in such reports is valuable in refining system requirements since they contain key information regarding the prevention of and recovery from future breaches of similar kind [2], [3]. Best practices in the past described in these textual artifacts can be considered as suggestions for future practices and a great supplement to legal requirements. This knowledge can be extracted by finding the common flow of historical event occurrences.

Our goal is to extract security and privacy requirements from such natural language artifacts presented as short stories. Understanding stories and narratives is a challenging task. The difficulties lie in the complexity of natural language and general knowledge of the world. Research on narrative events structures has gained increasing interest especially with the significant progress made in natural language processing (NLP). A large body of work focuses on commonsense inference by trying to understand the ground truth about the way the world works, which is difficult when models are trained on selected datasets, such as news articles. We focus on learning the common sense within the dataset itself. Analysis on repetitive and unsurprising stories, such as the aforementioned breach reports, can be valuable in itself without trying to understand the whole world.

One evaluation for understanding the structure of narrative events is the narrative cloze task [4], in which trained models infer held-out events based on extracted event sequences. This test requires formal representation of events (e.g., tuples of verb and arguments) and the event inference systems may rely on automatic linguistic preprocessing, such as part-of-speech tagging. The Story Cloze Test [5] removes this implicit requirement of NLP tools. The task is to generate a natural language ending to a preceding sequence of sentences to complete a story. To address these tests, models should be able to learn the common progression of events in the training set. Suitable models therefore can learn from previous events, which is nontrivial in the analysis of important texts such as breach reports.

Recent work has demonstrated that recurrent neural networks (RNNs) with Long Short-Term Memory (LSTM) [6] architectures are powerful models for complex tasks on sequential data, such as online handwriting recognition [7], speech recognition [8], and translation [9]. LSTM networks have been shown to have superior performance in the inference of events, as tuples [10] or raw tokens [11] in event chains. We evaluate their performance in inference of embedded vectors that represent sentences in a chain of vectors. For sentence embedding, we adopt Paragraph Vector (PV) [12], also known as Doc2Vec, which has been proven to have excellent performance in representing the meanings of documents, and achieves state-of-the-art results in text classification, information retrieval, and sentiment analysis [13], [14].

We aim at the task of inferring the held-out event, in the form of a sentence, given a sequence of context sentences in a story. By embedding sentences into vectors and capturing the progression of these vectors using LSTM networks, we

978-1-5386-8404-7/18/$31.00 ©2018 IEEE
DOI 10.1109/AIRE.2018.00009

intend to find common semantic shifts in stories. The proposed model produces the most probable vector for the held-out sentence, which we use to rank all possible answers by their cosine similarity to it. To evaluate the prediction model, we measure its performance on two datasets. Our results show that it significantly outperforms the baseline of using average word vectors.

The following is a sample breach report from the HHS breach report dataset:

> Two laptop computers with questionable encryption (each containing the electronic protected health information (ePHI) of 350,000 individuals) were stolen from the covered entity's (CE) premises. The types of ePHI involved included demographic and clinical information, diagnoses/conditions, medications, lab results, and other treatment data. After discovering the breach, the CE reported the theft to law enforcement and worked with the local police to recover the laptops. As a result of OCR's investigation, the CE developed and implemented new policies and procedures to comply with the Security Rule. The CE also provided breach notification to all affected individuals, HHS, and the media and placed an accounting of disclosures in the medical records of all affected individuals.

Previous studies have extracted and used security requirements extracted from this kind of reports [15], [16]. However, the sequential or causal relations between the requirements and events in the breaches were missing. In this work, we endeavor to find these missing relations. We convert breach reports into short stories, adapt our learning models accordingly, and then infer possible recovery and prevention sentences that could follow a breach description.

**Contributions** (i) A prediction model with PV and LSTM models for event inference in short stories, which has been evaluated on two different datasets; (ii) an automated process for curating HHS breach reports into structured short stories suitable for event inference; and (iii) an application of the proposed model on the analysis of breach reports.

**Structure** Section II reviews relevant background and related work. Section III presents the details of the prediction model. Section IV describes our evaluation process as well as the details of the two datasets. Section V demonstrates our results. Section VI discusses the significance and limitations of the model. Section VII describes potential future directions.

## II. Background and Related Work

We now introduce the techniques we incorporate in our method and related research on event inference.

### A. Background

**Word Embedding** Sentence embedding refers to the process of converting word sequences into vectors of real numbers, which is usually based on vectors of words from word embedding techniques. Word2Vec [17] is a celebrated model for this task resulting in word vectors of excellent quality and great scales [18]. Naively averaging word vectors as sentence vectors has been shown to perform well for tasks based on similarity and entailment [19], which we adopt as a baseline.

**Sentence Embedding** Sentence embedding based on word embedding techniques is a well studied, e.g., [18], [20], [21]. We adopt the Paragraph Vector method [12] for sentence embedding. As in Word2Vec, the sentence vectors are tasked to predict words given contexts from sentences, and can be considered as representations of their semantic meanings or topics. Our method seeks to capture the patterns of how these topics change in stories.

**RNN and LSTM Networks** Recurrent Neural Networks (RNNs) are neural networks with cycles in their computational graphs. They are recurrent because they perform the same task for data in a sequence, and have a short memory of what they have learned so far. The more complicated Long Short-Term Memory (LSTM) RNN networks [6] can "remember" short-term memories for a long period of time, and give excellent performances for sequence-to-sequence tasks, including a variety of NLP tasks [9], [22], [23].

**Event Inference** Structured sequences of participants and events, or scripts [24], are stereotypical sequences of structured events. Scripts are difficult to learn automatically because they do not adequately represent the semantics of events. Chambers and Jurafsky [4] introduced the concept of narrative event chains that represent structured knowledge to facilitate learning and enable inference. They also proposed the narrative cloze task, which is the inference of the event that is removed from a sequence of narrative events. In related studies, events are represented as tuples of verbs and dependencies [25] or arguments [26]. Mostafazadeh et al. [5] have introduced the Story Cloze Test [5] that lifts the requirements on formal representations of events, and tasks models to infer natural language endings to a rich set of everyday life stories. They also published the ROCStories corpus, a collection five-sentence commonsense stories, to enable this test.

### B. Related Work

The task of extracting usable information, such as formatted security requirements, from related textual artifacts is of great importance. Researchers start from designing and proposing systematic methodologies for manual extraction. Breaux et al. [27] have developed a methodology for manually extracting formal descriptions of rules, such as rights and obligations, that govern information systems from regulatory texts. They represent results from a case study on the text of HIPAA Privacy Rule. Hashmi [28] present a methodology for the extraction of legal norms from regulatory documents that emphasizes logical structures for reasoning and modeling to facilitate compliance checking. Automating the extracting process has been a challenge. Slankas et al. [29] propose an automated process for extracting access control policies implicitly and explicitly defined in natural language project artifacts. In these studies, requirements are usually stated explicitly in the text. In our study, however, we try to propose security requirements based on previous actions taken by responsible parties in similar scenarios, which is a new challenge.

Previous research has been conducted on event inference in stories. Kiros et al. [30] introduce the "skip-thought vectors" model for sentence representation using embeddings in low-dimensional space and RNNs. They test the effectiveness of their model for tasks including the prediction of neighboring

sentences in a corpus. We focus on sentence prediction and examine the implications of the predictions. Granroth-Wilding and Clark [31] present a neural network model to predict whether two events should appear in the same chain. They incorporate Word2Vec for the embedding of events. However, their work is limited to pair-wise association of events. Pichotta and Mooney [11] examine the effectiveness of LSTM language models using raw tokens of events with the narrative cloze evaluation. They conclude that sentence predictions based on raw tokens have comparable performances to systems using verb-argument events. We consider the insights from these studies in our proposed prediction model, and combine the event inference task with requirements engineering.

## III. Method

The goal of our study on event inference is to find the sequential semantic patterns, or narrative structures, among stories. We assume that some stories share similar semantic structures. For example, stories describing "playing a game" often end with "winning" or "losing" the game. We target short and repetitive stories where common patterns can be easily discovered and represented. These patterns can be valuable—they point toward common knowledge or norms that reside in textual documents. For example, in incident reports, sequential patterns like "laptop being stolen," "filing a police report," and "replacing alarm" provide insights to prevent and recover from future incidents.

To achieve this goal, we propose a method to infer held-out events in stories. A query to this task is a sequence of $N$ sentences ($s_1$, $s_2$, $\ldots s_N$), i.e., the context of a story. Our method is expected to predict a sentence, the ending or the leading one, that has been held-out from the story. Figure 1 illustrates our method.

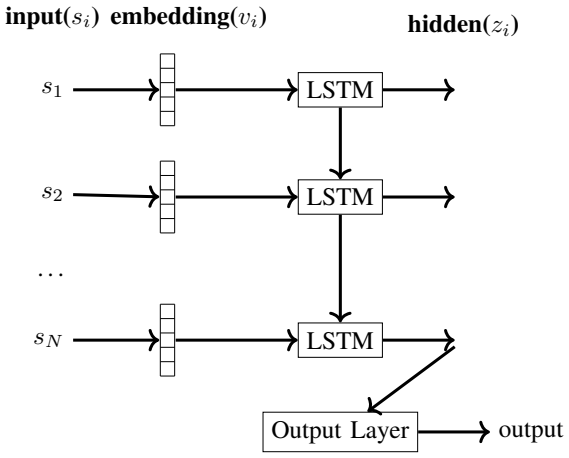


Fig. 1. Model to predict the vector of the held-out event.

**Baseline** As a baseline, we calculate the average word vector, learned with Word2Vec, of all words in the context. We also calculate the average word vectors of all target sentences. We rank all possible target sentences based on their cosine similarity to the average vector of the context.

**Sentence Embedding** We first embed all sentences into vectors using Paragraph Vector, which we normalize to unit length. Words in the text are stemmed beforehand, and only those appearing in at least two stories are kept.

**Inference Model** Sentence vectors are fed into our inference model, which comprises a number of LSTM models, the last of which produces a predicted vector for the target sentence through an output layer of neural network. Specifically, the output vector is:

$$o = normalize(tanh(W * z_N + B)),$$

where $z_N$ represents the hidden layers of the $N$-th LSTM cell. Weight matrix $W$ and the bias vector $B$ are randomly initialized. Leveraging a training set, the model learns and updates $W$ and $B$ by maximizing cosine similarity between the prediction and the correct answer, i.e., minimizing the cross entropy, defined as:

$$cross\_entropy = -\sum_{i=1}^{D} o_i * o_{i'},$$

where $D$ is the size of the vector space and $o'$ represents the evidence output. Note that $o$ and $o'$ are both normalized to unit length.

To interpret the output vector, we rank all possible answers from the corpus by their cosine similarity to this vector. The reciprocal of the rank of the correct answer is reported as a measure for the model's performance.

**Event Chain Prediction** Our method takes $N$ sentences and produces the prediction for the held-out sentence, e.g., the next sentence in the story. We can repeat this process and obtain a chain of events. The prediction of such event chains is useful in that they represent a common progression of stories, which can be interpreted as lessons learned from the past, based on the nature of the training data. To accomplish this task, we train multiple models that take different numbers of inputs. For example, the $i$th model takes $i$ input vectors and predicts the $(i+1)$st vector. The $i$ inputs combined with this predicted output are the input for the $(i+1)$st model. This process has some limitations, which we discuss in Section VI.

## IV. Evaluation

We evaluate our method on two datasets of short stories and report the prediction performance. Each short story we use is composed of five sentences. In our tests, we hold out one sentence, e.g., the ending, and use the others to predict it.

### A. Datasets: ROCStories

We adopt the ROCStories Winter 2017 release [32] corpus which includes 52,665 stories. The first four sentences are input and the fifth one is the output. In each of our tests, we use 90% (47,400) stories for training a model, and evaluate the model on the remaining 10% (5,265) stories.

Listing 1 shows an example from the ROCStories dataset.

Listing 1. Sample short story from the ROCStories corpus.

```
1 Tonight I played 3 games of online speed chess with Jim.
2 During the first game, the board froze.
3 Jim was able to checkmate me.
4 I signed off and went back on.
5 The board was ok this time, and I won the next game.
```

### B. Datasets: HHS Breach Reports

HHS breach reports are natural language texts that describe what happened in breaches where protected health information (PHI) was missing, improperly used, or impermissibly disclosed. In addition, they describe the actions that the covered entity (CE) or other relevant parties, such as Office for Civil Rights (OCR), have taken to mitigate the harms of breaches as well as to prevent similar breaches from recurring.

HHS is legally required to publish reports of breaches of unsecured PHI affecting 500 or more individuals. As of mid 2017, approximately 1,000 breach reports can be found on the HHS website, of length of six sentences on average. The reports usually start by describing the incident, and the types of PHI involved. The reports then list the events that happened after the breaches. Responsible parties are legally required to notify HHS, affected individuals, and the media, so the notification actions are mentioned in almost all reports.

To create a dataset of short stories from breach reports, we performed the following steps:

1) Removed breach reports that describe the same incidents.
2) Unified common synonyms, e.g., converting all mentions of "covered entity" to "CE."
3) Removed numbers, dates, and names.
4) Split sentences into smaller ones based on verbs using the NLP toolkit NLTK (http://www.nltk.org/). For example, the sentence "the CE reported the theft to law enforcement and worked with the local police to recover the laptops" is split into two sentences, namely, "the CE reported the theft to law enforcement" and "the CE worked with the local police to recover the laptops."
5) Removed sentences describing PHI types, notification, and trivial actions, such as "OCR obtained assurances that the CE implemented the corrective actions noted above."
6) Identified breach description sentences using heuristics, marked them, and numbered the sentences that describe the follow-up actions of the responsible parties.
7) Excluded breach reports that were either too short (fewer than five follow-up sentences) or too long (more than 10 follow-up sentences) after the above processes. We chose these numbers empirically based on the breach reporting styles in the dataset.

All steps expect step 1 were conducted automatically using NLP tools and heuristics. We also manually verified the results to reduce noise for the prediction task. Our resulting dataset contains 420 breach stories, including 420 breach descriptions and 2,182 follow-up sentences. This dataset is available here: https://goo.gl/aHcQRH.

Listing 2 shows the short story resulting from the seven-step process applied on the sample breach report in Section I.

Listing 2. Short story generated from the sample HHS breach report.

```
1 (Description) Two laptop computers with questionable
  encryption were stolen from the CE's premises.
2 (Follow-up 1) The CE reported the theft to law
  enforcement.
3 (Follow-up 2) The CE worked with the local police to
  recover the laptops.
4 (Follow-up 3) The CE developed and implemented new
  policies and procedures to comply with the Security Rule.
5 (Follow-up 4) The CE placed an accounting of disclosures
  in the medical records of all affected individuals.
```

Breach stories in our dataset consist of different numbers of sentences. The follow-up events correspond to the breach descriptions, but do not necessarily present causal ordering. For example, the fifth sentences of all breach stories are not usually their endings. To evaluate our model on held-out event prediction, we hold out the breach description, and use the other sentences to infer it. In our experiments, we only used the first four follow-up sentences of each breach story as inputs.

The more practical task to perform on breach reports is to infer follow-up sentences based on a breach description. As described in Section III, we build multiple models that accept sequences of different lengths. Specifically, the first model takes a breach description as input and predicts the first follow-up sentence; the second model takes the description and the first follow-up sentence, and predicts the second one, and so forth. Thus, these models can be combined to predict a chain of follow-up sentences based on the breach description.

### C. Metrics

For the prediction of held-out events, we require the models to rank all possible answers based on their probabilities of being the correct ones. Our method and the baseline rank all possible answers based on their similarity to the predicted vector. We measure the mean reciprocal rank (MRR) to evaluate their performances:

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i},$$

where $|Q|$ is the size of the test set (5,265 for the ROCStories dataset and 42 for breach reports dataset), and $rank_i$ is the rank position of the correct answer for the $i$-th query. A better model will give higher ranks (smaller numbers) to the correct answers and will therefore have a higher MRR value. It is easy to calculate that, if the maximum rank position is $N$, the average MRR for a random selection is proximately $\frac{ln(N)}{N-1}$. Note that the maximum rank positions are 52,665 and 420 for ROCStories and breach reports datasets, respectively.

In addition, we calculate Top-1% ratios of the results, which are the percentages of the instances in which the correct answers are ranked in the top 1% of all possible answers.

## V. Results

We used the Deeplearning4j (https://deeplearning4j.org/) library to perform the Paragraph Vector method and converted sentences into 100-dimensional vectors. We used TensorFlow (https://www.tensorflow.org/) to build 500-dimensional LSTM cells for our model. When we applied the PV to the breach

reports, the description of a breach, which may contain multiple sentences, is given one label. Thus, each story has 1+4 sentence vectors in the tests.

### A. Held-Out Events Prediction

For each query, our method produces a list of sentences from all possible answers, ranked based on their cosine similarity to the predicted vector. For the ROCStories dataset, the set of possible answers is all of the 52,665 fifth sentences. For the breach reports dataset, it is all of the 420 breach descriptions. We measured and compared the MRRs and Top-1% ratios of our method and the baseline. We added the results from a random selection as scales to the numbers. Based on the selection of training sets and random initial assignments of the weights in the network, the results may vary. We ran each test three times and report the average numbers in Table I.

TABLE I
RESULTS OF HELD-OUT EVENT PREDICTIONS.

| **Datasets** | | ROCStories | HHS Breach Reports |
|---|---|---|---|
| **Our Model** | MRR | **0.0685** | **0.0487** |
| | Top-1% | **45.2%** | **21.4%** |
| **Average Word2Vec** | MRR | 0.0395 | 0.0155 |
| | Top-1% | 29.8% | 11.9% |
| **Random** | MRR | 0.0002 | 0.0118 |
| | Top-1% | 1.0% | 1.0% |

Table II shows the top five possible ending sentences for Listing 1, which is not present in the training set.

TABLE II
LIST OF SENTENCES RETRIEVED AS THE ENDING OF LISTING 1.

| **Rank** | **Sim** | **Sentence** |
|---|---|---|
| 1 | 0.604 | I won the case, and the debt was dropped. |
| 2 | 0.560 | The board was ok this time, and I won the next game. **(correct answer)** |
| 3 | 0.548 | When the game was close to ending I won! |
| 4 | 0.545 | Our team won the game. |
| 5 | 0.540 | I signed off after my lucky game. |

Using the four follow-up sentences in Listing 2 as input, the top three predicted breach descriptions are listed in Table III, which are all highly similar to the actual description. This story is not present in the training set, either.

### B. Follow-up Events Prediction

In our tests, we trained four models to predict four follow-up vectors based on a breach description. For each vector, we found the sentence with the highest cosine similarity to it, out of the 2,182 follow-up sentences in the database, as the eventual result. Table IV shows an example result for the breach description in Listing 2. Our models generated a sequence of four vectors to follow the breach description, and the most similar (top-1 ranked) sentence to each of them is listed. We also list the actual follow-up sentences in the report along with their similarities to the predictions and their ranks. We consider these predicted sentences as requirements to recover from or prevent similar breaches.

We manually examined all predicted follow-up requirements for a testing set, and found that 60% of the predictions were plausible and 35% matched what was reported.

TABLE III
LIST OF BREACH DESCRIPTIONS RETRIEVED BASED ON FOLLOW-UP SENTENCES IN LISTING 2.

| **Rank** | **Sim** | **Sentence** |
|---|---|---|
| 1 | 0.912 | A former employee of the CE retained possession of a retired unencrypted laptop computer that contained PHI following his termination. |
| 2 | 0.909 | Unencrypted laptop computer containing the PHI of patients was stolen from the CE's administrative offices during a break-in. |
| 3 | 0.907 | The CE reported that an unencrypted laptop computer that contained the ePHI of patients was stolen from a clinic. |

TABLE IV
PREDICTED REQUIREMENTS FOR THE BREACH DESCRIPTION IN LISTING 2.

| **Rank** | **Sim** | **Sentence** |
|---|---|---|
| | | (Input) Two laptop computers with questionable encryption of individuals were stolen from the CE's premises. |
| 1 | 0.867 | (Predicted requirement 1) The CE filed a police report to recover the stolen item. |
| 1 | 0.795 | (Predicted requirement 2) The CE replaced its building alarm and installed bars on the windows. |
| 1 | 0.869 | (Predicted requirement 3) The CE revised its existing policies to ensure its vendors enforce appropriate security measures to protect ePHI. |
| 1 | 0.873 | (Predicted requirement 4) The CE implemented mandatory encryption for all mobile devices. |
| 55 | 0.653 | (Follow-up 1) The CE reported the theft to law enforcement. |
| 1314 | 0.356 | (Follow-up 2) The CE worked with the local police to recover the laptops. |
| 89 | 0.743 | (Follow-up 3) The CE developed and implemented new policies and procedures to comply with the Security Rule. |
| 1896 | 0.034 | (Follow-up 4) The CE placed an accounting of disclosures in the medical records of all affected individuals. |

## VI. DISCUSSION

We first discuss the significance of our results, and then review the limitations of our method.

### A. Significance

**Common flows of stories** Our method is effective in capturing the common semantic shifts in stories. If the stories are short and repetitive, common patterns can be mined from them. It is worth noticing that, for the ROCStories corpus, Average Word2Vec works relatively well for held-out predictions. Approximately 30% of the correct answers are ranked in top 1% in the results. Sentences in one story often contain similar words. For example, consider the following story:

*"Molly is out swimming. Molly swims deep into the ocean. Molly feels something swim near her. A shark begins to attack molly. Molly swims back to shore and survives the attack."*

It is not surprising that this story ends with a sentence containing "Molly" and "swim." Our method, however, can predict more diverse endings by capturing the common flow of events. For example, it was able to infer that I will "start my day" after I "wake up," "make coffee," and "take a shower."

We conducted evaluations of held-out events prediction on HHS breach reports because breach descriptions could be semantically different from the follow-up sentences. For this task, Average Word2Vec gives results closer to a random selection, while our method performs better, but the results were worse than those of the tests on ROCStories.

**Requirement suggestions** From a practical standpoint, predicting follow-up events based on a breach description is more valuable than predicting held-out descriptions from follow-up events. This task is helpful to the analysis of breach reports since it produces actions that are most probable to follow a certain breach. We can consider these predicted actions as security requirements, since responsible parties can follow such actions to recover from and prevent similar breaches. For example, our prediction results in Table IV suggest "mandatory encryption for all mobile devices" which is a pertinent suggestion regarding this breach description but was missing in the original report. When giving such suggestions in practice, we can additionally present previous breaches in which the listed actions were taken as validation of the suggestions.

Instead of unstructured sentences, we also can produce formalized requirements by applying our previous framework for norm extraction [15] to these results. For example, from the predicted sentences in Table III, security requirements in the form of norms can be extracted, as listed in Table V. A norm in the particular sense we adopt here is a directed relationship between two parties that regulates the interaction among the parties [33]. We consider three types of norms: commitments, authorizations, and prohibitions. A commitment type of norm, written as $c$(SUBJECT, OBJECT, antecedent, consequent), represents that the subject commits to the object to bring about the consequent when the antecedent holds.

TABLE V
FORMALIZATION OF EXTRACTED REQUIREMENTS IN TABLE III AS NORMS

| |
|---|
| (Predicted requirement 1) The CE filed a police report to recover the stolen item.<br>- $c$(CE, NONE, a theft has happened, notify law enforcement) |
| (Predicted requirement 2) The CE replaced its building alarm and installed bars on the windows.<br>- $c$(CE, NONE, TRUE, implement safeguard on physical workspace)<br>- $c$(CE, NONE, workstations contain PHI, limit access to workstations) |
| (Predicted requirement 3) The CE revised its existing policies to ensure its vendors enforce appropriate security measures to protect ePHI.<br>- $c$(CE, NONE, working with subcontractors, obtain proper agreements on data security) |
| (Predicted requirement 4) The CE implemented mandatory encryption for all mobile devices.<br>- $c$(CE, NONE, portable devices contain PHI, encrypt portable devices) |

### B. Limitations

Our method performs better than the baseline, but the results may not be considered as strong. Our future work includes improving our prediction model. Additionally, our method suffers from the following limitations.

**Paragraph Vector** The Paragraph Vector method for sentence embedding, albeit suitable for capturing semantic similarity among sentences, does not preserve all information that a sentence conveys, especially for compound sentences. This fact limits our method on the analysis of stories with short and simple sentences. Breach descriptions are usually longer and contain some key information that affects the follow-up events. Sentence embeddings that leverage importance of words, such as attention-based embeddings [20], could be more effective. For example, the phrase "laptops were stolen" may be more important than "the CE reported" when they appear in the same sentence.

**Causal relationships** Our model only learns the sequential ordering of similar sentences in the training set, and does not examine the causal relationships among the events and their actors. The performance of our method is sensitive to the representativeness of the training set. It is unable to infer surprising events or endings, as in the following story:
*"The bride smiled at the groom. She was missing a tooth. The groom thought that was cute. He remembered that moment."*

Our model produces the sentence "She screamed and said yes" as an ending, which could be semantically relevant, while the provided ending is "He shared it at her funeral," which has no similar precedents in the training set, but makes sense to a human reader. A more effective model should be able to offer highly probable predictions with more diversity.

**Limitations of the dataset** We have found that predictions are more accurate when there are similar stories to the queries in the dataset. Although the HHS breach reports include recurrence of similar incidents, the dataset has a limited size. The performance of the predictions presented noticeable variations in our tests. Manual examination and heuristics may be more suitable for its analysis for now. Also, we assume that the reports include only "best practices" which is not necessarily true. We call for more attention on the usefulness

of breach reports and hope that more reports of high quality will be documented. We plan to keep our dataset updated at the meantime.

**Overfitting** With a relatively small dataset such as the set of breach reports, our model may be overfitting. In our experiments, the model's performance on testing datasets could sometimes decline with more iterations for training. We mitigated this problem by limiting the number of iterations. Regularization could be a more reliable solution to this problem, which we leave to future work.

**Follow-up events prediction** We infer multiple follow-up events based on one breach description. As a result, our model produces similar or identical event predictions for similar breach descriptions, which is expected, whereas the actual follow-up events may be more tailored to the details of the description. Breach descriptions tend to be long sentences with detailed information that may not have been accurately or completely captured by the sentence vectors. For example, some actions that a CE might take depend on whether a business associate (BA) was involved in the breach. Breach stories can differ considerably between "CE losing laptops" and "BA of CE losing laptops", even though the breach descriptions may be semantically similar. To make the predictions more effective, we can incorporate richer features from the breach descriptions, including characters based on heuristics. We leave this to future work.

## VII. Conclusions and Future Work

We have proposed a prediction model for the inference of held-out events in stories. It first converts sentences into vectors using the Paragraph Vector method; adopts LSTM networks to predict a vector for the held-out event; and selects sentences most similar to this vector. We have demonstrated that the proposed model outperforms the baseline of using average word vectors. Event inference based on breach descriptions can suggest useful requirements for security practitioners and end users to take after a breach has happened. Future work includes experimenting on different document embedding methods, extracting and incorporating further information in sentences such as heuristic features, learning causal relationships among events from text, and exploring other suitable corpora, specially security related story-like textual artifacts.

## References


[1] HHS Breach Portal, "Notice to the Secretary of HHS breach of unsecured protected health information affecting 500 or more individuals," 2016, United States Department of Health and Human Services (HHS). https://ocrportal.hhs.gov/ocr/breach/.

[2] Y. Liu, A. Sarabi, J. Zhang, P. Naghizadeh, M. Karir, M. Bailey, and M. Liu, "Cloudy with a chance of breach: Forecasting cyber security incidents," in *Proc. USENIX Conference on Security Symposium*, 2015, pp. 1009–1024.

[3] M. Riaz, J. Stallings, M. P. Singh, J. Slankas, and L. Williams, "DIGS: A framework for discovering goals for security requirements engineering," in *Proc. ESEM*. ACM, 2016, pp. 35:1–35:10.

[4] N. Chambers and D. Jurafsky, "Unsupervised learning of narrative event chains," in *ACL*, K. McKeown, J. D. Moore, S. Teufel, J. Allan, and S. Furui, 2008, pp. 789–797.

[5] N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. Allen, "A corpus and cloze evaluation for deeper understanding of commonsense stories," in *Proc. NAAACL-HLT*, 2016, pp. 839–849.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, 1997, pp. 1735–1780.

[7] A. Graves, S. Fernández, M. Liwicki, H. Bunke, and J. Schmidhuber, "Unconstrained online handwriting recognition with recurrent neural networks," in *Proc. NIPS*, 2007, pp. 577–584.

[8] A. Graves, A. rahman Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," *ICASSP*, 2013, pp. 6645–6649.

[9] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. NIPS*, 2014, pp. 3104–3112.

[10] K. Pichotta and R. J. Mooney, "Learning statistical scripts with lstm recurrent neural networks," in *Proc. AAAI*, 2016, pp. 2800–2806.

[11] K. Pichotta and R. J. Mooney, "Using sentence-level LSTM language models for script inference," in *Proc. ACL*, 2016.

[12] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. ICML*, 2014, pp. 1188–1196.

[13] A. M. Dai, C. Olah, Q. V. Le, and G. S. Corrado, "Document embedding with paragraph vectors," in *NIPS Deep Learning and Representation Learning Workshop*, 2014.

[14] Q. Ai, L. Yang, J. Guo, and W. B. Croft, "Analysis of the paragraph vector model for information retrieval," in *Proc. ICTIR*, 2016, pp. 133–142.

[15] H. Guo, O. Kafalı, A.-L. Jeukeng, L. Williams, and M. P. Singh, "Toward extraction of security requirements from text: Poster," in *Proc. HoTSoS*, 2018, pp. 27:1–27:1.

[16] Ö. Kafalı, J. Jones, M. Petruso, L. Williams, and M. P. Singh, "How good is a security policy against real breaches? a HIPAA case study," in *Proc. ICSE*, 2017, pp. 530–540.

[17] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. NIPS*, 2013, pp. 3111–3119.

[18] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger, "From word embeddings to document distances," in *Proc. ICML*, 2015, pp. 957–966.

[19] J. Wieting, M. Bansal, K. Gimpel, and K. Livescu, "Towards universal paraphrastic sentence embeddings," in *Proc. ICLR*, 2016.

[20] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based lstm for aspect-level sentiment classification," in *EMNLP*, 2016, pp. 606–615.

[21] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," in *Proc. ICLR*, 2017.

[22] O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton, "Grammar as a foreign language," in *Proc. NIPS*, 2015, pp. 2773–2781.

[23] K. M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Proc. NIPS*, 2015, pp. 1693–1701.

[24] R. C. Schank and R. P. Abelson, *Scripts, plans, goals and understanding: An inquiry into human knowledge structures*. Oxford, England: Lawrence Erlbaum, 1977.

[25] R. Rudinger, P. Rastogi, F. Ferraro, and B. Van Durme, "Script induction as language modeling," in *Proc. EMNLP*, 2015, pp. 1681–1686.

[26] K. Pichotta and R. J. Mooney, "Statistical script learning with multi-argument events," in *Proc. EACL 2014*, 2014, pp. 220–229.

[27] T. D. Breaux and A. I. Antón, "Analyzing regulatory rules for privacy and security requirements," *IEEE Transactions on Software Engineering*, vol. 34, no. 1, pp. 5–20, 2008.

[28] M. Hashmi, "A methodology for extracting legal norms from regulatory documents," in *Proc. IEEE EDOCW*, 2015, pp. 41–50.

[29] J. Slankas and L. Williams, "Access control policy extraction from unconstrained natural language text," in *Proc. SocialCom*, 2013, pp. 435–440.

[30] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler, "Skip-thought vectors," in *Proc. NIPS*, 2015, pp. 3294–3302.

[31] M. Granroth-Wilding and S. Clark, "What happens next? event prediction using a compositional neural network model," in *Proc. AAAI*, 2016, pp. 2727–2733.

[32] N. Mostafazadeh, M. Roth, A. Louis, N. Chambers, and J. Allen, "Lsdsem 2017 shared task: The story cloze test," in *Proc. the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, 2017, pp. 46–51.

[33] M. P. Singh, "Norms as a basis for governing sociotechnical systems," *ACM TIST*, vol. 5, no. 1, pp. 21:1–21:23, 2013.