Document Sensitivity Classification for Data Leakage Prevention with Twitter-based Document Embedding and Query Expansion

Lap Q. Trieu, Trung-Nguyen Tran, Mai-Khiem Tran, Minh-Triet Tran
Faculty of Information Technology
University of Science, VNU-HCM
Ho Chi Minh City, Vietnam
tqlap@apcs.vn, ttnguyen@selab.hcmus.edu.vn,
tmkhiem@selab.hcmus.edu.vn, tmtriet@fit.hcmus.edu.vn

***Abstract*—Document sensitivity classification is essential to prevent potential sensitive data leakage for individuals and organizations. As most of existing methods use regular expressions or data fingerprinting to classify sensitive documents, they may not fully exploit the semantic and content of a document, especially with informal messages and files. This motivates the authors to propose a novel method to classify document sensitivity in realtime with better semantic and content analysis.**

Taking advantages of deep learning in natural language processing, we use our pre-trained Twitter-based document embedding TD2V to encode a document or a text fragment into a fixed length vector of 300 dimensions. Then we use retrieval and automatic query expansion to retrieve a re-ranked list of semantically similar known documents, and determine the sensitivity score for a new document from those of the retrieved documents in this list. Experimental results show that our method can achieve classification accuracy of more than 99.9% for 4 datasets (snowden, Mormon, Dyncorp, TM) and 98.34% for Enron dataset. Furthermore, our method can early predict a sensitive document from a short text fragment with the accuracy higher than 98.84%.

***Keywords*-Sensitive document detection, document embedding, Doc2Vec, automatic query expansion, data leakage prevention**

I. Introduction

Data leakage prevention (DLP[1]) is one of the essential problems to protect personal and organizational sensitive information from disclosing without official consent. The number of confidential data leaks is increasing every year [2]. In Global Data Leakage Report of InfoWatch Analytical Center, there are 1,556 confidential data leaks registered in 2016, increasing 3.4% more than in 2015[3]. Especially, more than 3.1 billion personal data records are compromised, up to three times more than in 2015[3].

With the increasing amount of data created everyday, it is not an easy task for people to classify and manage documents based on their sensitivity levels. Currently most of existing methods in DLP use regular expressions or data fingerprinting[4]. Thus these methods only analyze the formal representation, i.e. data format, not the semantic content of a document[5]. This approach is appropriate with known patterns and templates of a classified document, such as contracts or agreements, but may not efficiently help people to aware the potential privacy leaks in informal documents, such as emails or personal notes.

In recent years, natural language processing (NLP) techniques, such as N-gram[4] or Named Entity Recognition[6], have been applied for document classification in DLP. In this paper, we follow this trend to better understand and exploit document content to classify document sensitivity. Inpired by ParagraphVector [7], one of the state-of-the-art methods for document representation, we apply our Twitter-based Doc2Vec model (TD2V [8]) to vectorize an arbitrary document or text fragment. Collecting more than one million tweets (from 2010 to 2017) in Twitters, we train TD2V from 422,351 English articles with 297,298,525 tokenized words [8]. Thus, our word embedding model is expected to be general and efficient enough to represent English documents and text fragments in various domains. Then, we propose a novel method to classify the sensitivity level of a document d using retrieval and automatic query expansion (AQE [9]). From the initial ranklist containing k nearest neighbors of d, we use Modified Distance[8] to re-rank documents in a labeled sensitivity corpus S. The sensitivity label for d is determined by majority voting scheme from the top l in this re-ranked list.

Following the work by Hart et.al. [10], we gather four different datasets, namely, *Dyncorp*, *Transcendental meditation* (TM), *Mormon*, and *Enron*. We also create the fifth dataset *Snowden*. Our experiment for full document classification shows that our proposed method achieves the accuracy more than 99.9% for 4 datasets (*Snowden*, *Mormon*, *Dyncorp*, *TM*) and 98.34% for *Enron* dataset. Besides, we also conduct sensitivity classification for short text segment (512 bytes, 1KB, 2KB, and 4KB) and our method achieves the accuracy of more than 99.7% for 4 datasets (*Snowden*, *Mormon*, *Dyncorp*, *TM*) and 98.84% for 1 dataset (*Enron*).

The main contributions of our work are as follows.

- we propose a novel method to classify the sensitivity of a document or a text fragment with two phases: text fragment vectorization with our pre-trained document

embedding, and sensitivity classification with majority vote from a re-ranked list of similar labeled document using retrieval and query expansion. Our method can also early predicts a potentially sensitive document just from a short fragment.
- we develop a system to continuously monitor user's behaviors to detect activities that may leak potentially sensitive data via network and removable storage devices.

This paper is organized as follows. In section II, we briefly review existing methods and approaches for document and text classification for data leakage prevention, then we discuss the related work on semantic document representation that we follow to propose our new method in this paper. Section III presents our method to classify document sensitivity based on textual content. Section IV presents datasets and experimental results. Our proposed system to help users to detect potential activities that may leak sensitive data is discussed in Section V. Finally, the conclusion is in Section VI.

II. Related Work

A. Document or Text Classification for DLP

A common approach in recent DLP systems is content analysis. The content of stored or exchanged data and network traffic is analyzed to detect potential threat for sensitive data leakage [4]. Most of existing DLP methods use syntactic patterns to identify data sensitivity[2], [1]. Thus, they are appropriate to process documents with well-defined templates, but may not exploit the semantic in document content in emails, personal notes, or enterprise documents.

A new trend for DLP is based on natural language processing (NLP). D.Du et.al. propose a method using latent semantic analysis to extract semantic features representing concepts[5]. S.Alneyadi et.al. develop a method using N-gram based statistical analysis for DLP[4]. Named Entity Recognition is also used to check known entities in documents to prevent sensitive data leakage[6].

In this paper, we also follow the trend to apply NLP to analyse and represent the semantic of documents or text fragments for DLP. The brief review of common methods for document representation is presented in Section II-B.

B. Document Representation

A document is an ordered list of words and phrases with a flexible length. Therefore, a document or a text fragment should be encoded into a fixed length vector for further analysis, such as classification. A well-known approach to vectorize a document is Bag-of-Words (BoW) model in linguistic domain, first introduced by Z.Harris et.al. in 1954 [11]. Many techniques and variants have been proposed to better represent documents, such as Fuzzy Bag-of-Words[12].

As the word order is usually ignored in BoW methods, they may not fully capture the semantic relationships between words and the textual content of a document[7]. Distributed representations for words [13] by Rumelhart et. al. is considered as the most successful concept [14] using machine learning techniques to embed words in a document into a vector space while preserving the semantic meaning of a given context. This paradigm opens a new trend to use neural networks for word embedding[14], [7].

Inspired Paragraph Vector[7], proposed by Quoc Le et.al for distributed representation of sentences and documents, we train our own pre-trained work embedding model, namely TD2V (Twitter-based Doc2Vec[8]), from more than one million tweets in Twitter from 2010 to 2017. As this pre-trained model is created from a large corpus with 297,298,525 tokenized words from 422,351 English articles in various online sources, TD2V is expected to capture the semantic of English text fragments in many domains. In this paper, we propose a direct application of TD2V to encode from a short text fragment to a lengthy document into a fixed-length vector for sensitivity classification, and apply this module into our system for data leakage prevention.

III. Proposed Method

A. Method Overview

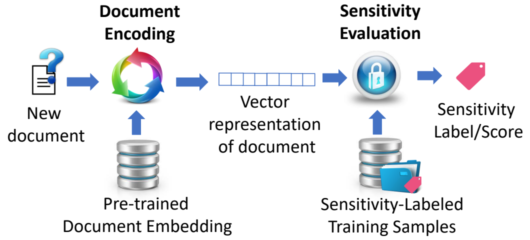
Document Encoding
Sensitivity Evaluation
New document
Vector representation of document
Sensitivity Label/Score
Pre-trained Document Embedding
Sensitivity-Labeled Training Samples

Figure 1. Overview of our proposed method to evaluate document sensitivity.

Figure 1 illustrates our proposed method to classify document sensitivity with two main phases:

- **Document Encoding**: a document or a short text fragment is encoded into a fixed-length vector with a pre-trained document embedding. In our method, we choose TD2V[8], a Twitter-based Doc2Vec model, to vectorize a document or a text fragment. This phase is presented in Section III-B.
- **Sensitivity Evaluation**: we restate the problem to evaluate sensitivity into a retrieval task, i.e. we retrieve a list of most semantically similar known samples from a sensitivity corpus S to determine the sensitivity label/score for a new document or text fragment. (c.f. Section III-C). Modified Distance[8] is used to better measure the semantic distance between two document vectors.

B. Document Representation with Twitter-based Document Embedding

We may need to train a dedicated document embedding model to encode a text fragment for sensitivity evaluation. However, this task needs a large enough corpus and consumes high computational cost. Thus, we decide to re-use our pre-trained Twitter-based model TD2V [8] to encode an arbitrary English text fragment. Initially proposed to encode documents for news article classification, TD2V can also be used to encode text fragment for sensitivity evaluation as it has been trained with a large enough and multi-domain corpus containing $n_D = 422,351$ articles with $297,298,525$ tokenized words from various online sources. Experiments in Section IV show that TD2V can be used to efficiently encode English documents/text fragments for sensitivity evaluation.

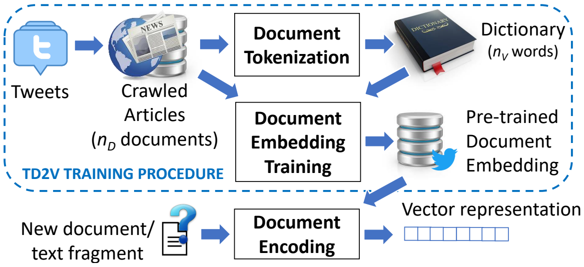
Tweets
Crawled Articles (n_D documents)
Document Tokenization
Dictionary (n_V words)
Document Embedding Training
Pre-trained Document Embedding
TD2V TRAINING PROCEDURE
New document/ text fragment
Document Encoding
Vector representation

Figure 2. Encode a document/text fragment with a pre-trained Twitter-based Document Embedding

Figure 2 shows the procedure to encode a document or text fragment using TD2V[8]. For details of training TD2V, please refer to [8]. Here, we only summarize the processes to train TD2V and to encode a new document using TD2V following distributed memory PV-DM[7] (Figure 3).

From tokenized words of all documents in the training corpus, a dictionary with $n_V = 271,620$ terms is defined after filtering the words with frequency lower than $n_{minFrequency} = 5$. The objective of the training process is to obtain word embedding W and document embedding D from the training corpus to maximize the probability to predict a target word given context words within a document context. In the network of this training process, the input vector (with $n_V + n_D$ elements) is the concatenation of context-word-IDs vector and document-ID vector, the hidden layer consists of $size$ nodes, and the output layer has n_V nodes to represent the predicted target word. Through experiments in [8], the length of document vector is selected to be $size = 300$.

To encode a new document d, as illustrated in Figure 3 (right), the document-ID vector in the input layer is replaced with a single value node d. The objective is to learn the $1 \times size$ matrix D while keeping the learned matrices W and W' fixed. D is used to represent the new document d.

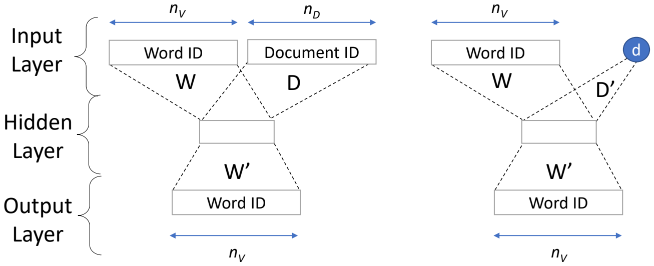

Figure 3. Train a document embedding model (left) and encode a new document with the learned model (right)[8]

C. Sensitivity Evaluation with Retrieval and Automatic Query Expansion

To determine the sensitivity label for a document or text fragment, we restate the problem into a retrieval task (Figure 4). For a given text fragment x, we retrieve a list of most similar samples in a sensitivity-labeled dataset S, then infer the sensitivity label or score of x.

Instead of using the Euclidean distance $d(x, y)$ between two documents x and y, we use Modified Distance [8] to better measure the relationship between documents. The key idea of this metric is to further exploit the external relationships from the k nearest neighbors of x to have better view about the context and semantic of x, not just its direct neighbors.

To evaluate the Modified Distance of x and other labeled samples, we first retrieve a ranklist $RL_k(x)$ with k most relevant samples in sensitivity corpus S, sorted in ascending order of Euclidean distance from x.

$$RL_k(x) = (y_{1,x}, y_{2,x}, ..., y_{k,x}) \tag{1}$$

where $y_{i,x} \in S$ and $d(x, y_{i,x}) \leq d(x, y_{i+1,x}), 1 \leq i < k$.

Modified Distance $\hat{d}(x, y)$ can now be defined as a weighted combination of direct distance $d(x, y)$ and the average direct distance from all k neighbors of x to y:

$$\hat{d}(q, x) = \alpha.d(q, x) + (1 - \alpha) \frac{\sum\limits_{z_i \in RL_k(x)} d(z_i, y)}{|RL_k(x)|} \tag{2}$$

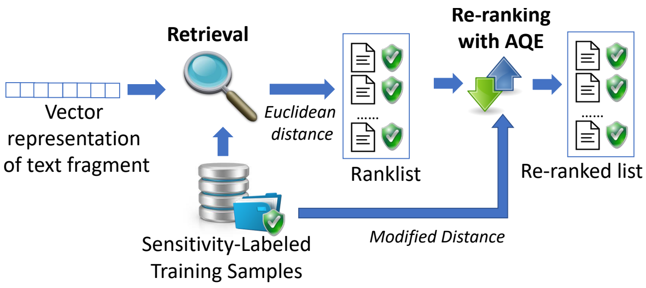

Figure 4. Retrieve relevant labeled samples from sensitivity training dataset

.

Based on the idea of automatic query expansion (AQE), we determine the re-ranked list $\widehat{RL}_{k,l}\left(x\right)$ with l samples in the sensitivity training corpus S using the ranklist $RL_k\left(x\right)$ using Modified Distance.

$$\widehat{RL}_{k,l}\left(x\right) = \left(\hat{y}_{1,x}, \hat{y}_{2,x}, ..., \hat{y}_{l,x}\right) \quad (3)$$

where $\hat{y}_{i,x} \in S$ and $\hat{d}\left(x, \hat{y}_{i,x}\right) \leq \hat{d}\left(x, \hat{y}_{i+1,x}\right), 1 \leq i < l$.

The sensitivity label for a text fragment/document x can now be determined using majority voting scheme from the labels of known training samples in the re-ranked list $\widehat{RL}_{k,l}\left(x\right)$. The label set can be {sensitive, insensitive}, or can have multiple levels of sensitivity. In the experiments in this paper, we consider the binary classification. In practice, we can define $\{s_0, s_1, ...s_{\max}\}$ as a set of sensitivity levels, where s_0 is insensitive, and $s_{\max}$ is the most sensitive level. The score for a document can be determined by the (weighted) average score of all training documents in the re-ranked list $\widehat{RL}_{k,l}\left(x\right)$. For all experiments in this paper, we choose $\alpha = 0.5$, $k = 5$ and $l = 7$.

IV. Experiments

A. Description of Datasets

For experiments, we gather four datasets mentioned in [10], namely, *Dyncorp*, *Transcendental meditation*, *Mormon*, and *Enron*. The first three datasets contain the combination of sensitive documents (private enterprise) from WikiLeaks and public enterprise documents collecting from its public websites, respectively. We also create own dataset *Snowden* with secret documents related to NSA's program. The details information of 5 datasets are as follows:

Dyncorp is a collection of military contractor documents with 25 sensitive documents and 362 public related documents from its website. Due to the difference between the sizes of sensitive documents and the purpose of generating enough textual information for classification, we further divide the two largest documents in this collection into 77 small files (˜2000 characters/file). In the end, there are totally 254 private documents for Dyncorp dataset.

Transcendental meditation is a religious organization including workshop instructions written by high-ranking members of the organization. Its collection contains 29 sensitive documents from WikiLeaks and 144 insensitive documents crawled from its public website based on our Twitter crawler. Similarly, the five largest sensitive documents in the dataset are segmented into 44 files using the same technique, increasing the total number of private documents of Transcendental meditation dataset to 68.

Mormon dataset is a Mormon handbook referenced from The Church of Jesus Christ of Latter-day Saints (LDS). We split the handbook data into ˜2000 character-long pieces, resulting in the collection of 274 private documents. We use 141 webpages from the Church of Jesus Christ of Latter Day Saints website crawled from LDSchurch Twitter channel as public enterprise documents for this dataset.

Enron collection contains emails released during the Federal Energy Regulatory Commission labeled by Hearst et al. Due to the size of this document (more than 300 MB), we split it into ˜5000 character-long files. Hence, *Enron* dataset contains totally 64,304 private documents. In this paper, we use only 10,000 files for evaluation.

Snowden dataset is a collection of 1025 secret documents relating to the National Security Agencys (NSAs) dragnet mass surveillance program during the period of five years from 2013-2017 of Edward Snowden.

For each dataset, we use the collection of 10,000 random wiki articles and more than 20 thousand of news articles collecting from different Twitter channels to represent for non-enterprise/insensitive data. Table I shows the number of sensitive and insensitive (enterprise and non-enterprise) documents in datasets.

B. Document Fragment Sensitivity Classification

In this experiment, we evaluate the accuracy of our method on document fragments of different sizes. Our goal is to verify if the proposed method can ***quickly recognize a potentially sensitive document*** just by scanning through a small part of the document.

We consider four values for fragment size: 512, 1024, 2048, and 4096 bytes. For each fragment size, to increase the number of sensitive documents for classification, every sensitive document is randomly segmented into 20 small files. Sensitive documents whose lengths are less than the current fragment size are not split but used the whole content for evaluation instead. For every insensitive document, we use only the first part of the content with the same length as the package size. As shown in Table II, corresponding

Table I
Number of sensitive and insensitive documents in datasets.

Dataset	Private enterprise (Sensitive)	Public enterprise (insensitive)	Non-enterprise (insensitive)
Dyncorp	254	362	22945 news + 10000 wiki articles
TM	68	144	
Enron	10000	0	
Mormon	274	141	
Snowden	1025	0	

Table II
Number of document fragments in datasets

Size	Sensitive	*Mormon*	*Dyncorp*	*TM*	*Enron*	*Snowden*
512	Yes	5480	5080	1341	10000	20500
	No	33086	33307	33089	32945	32945
1024	Yes	5480	5061	1284	10000	18657
	No	33086	33307	33089	32945	32945
2048	Yes	293	596	467	10000	14610
	No	33086	33307	33089	32945	32945
4096	Yes	274	482	258	10000	6714
	No	33086	33307	33089	32945	32945

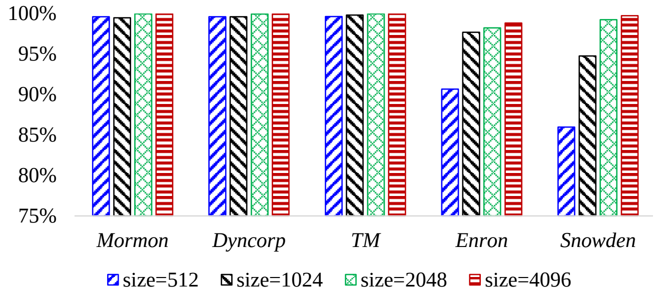

Figure 5. Classification accuracy for document fragments with different fragment sizes

to different fragment sizes, the numbers of sensitive and insensitive fragments in each dataset can be different due to the number of fragments that can be generated from the original data. We use 50% data for training, and 50% for testing.

Figure 5 visualizes the accuracy for sensitivity classification of document fragments with different fragment sizes. The details information are in Table III. For all datasets, the accuracy increases when the fragment size increases. For *Mormon*, *Dyncorp*, and *TM* datasets, the accuracy is higher than 99% even with very short size (512 bytes). For *Enron* and *Snowden* datasets, the accuracy significantly increases when the fragment size increases from 512 to 1024 bytes, and can be as high as 98.84% and 99.73% with fragment size of 4096 bytes. We can see that even with the very short size (512 bytes), our method can correctly classify document fragments with the accuracy higher than 90% (except for 85.98% with *Snowden* dataset).

Table III
PERCENTAGE OF SENSITIVITY CLASSIFICATION FOR DOCUMENT FRAGMENTS

Size	*Mormon*	*Dyncorp*	*TM*	*Enron*	*Snowden*
512	99.606	99.614	99.640	90.672	85.982
1024	99.481	99.614	99.825	97.699	94.768
2048	99.964	99.947	99.982	98.240	99.239
4096	99.970	99.953	99.976	98.840	99.734

We further analyse the number of misclassified sensitive fragments to see if our method can easily allow a sensitive fragment to be ignored (Table IV). For each dataset and a fragment size, we report the number of false negatives over the total number of sensitive fragments. From this table, we see that less than 2% of sensitive fragments might be misclassified as regular fragments (with fragment size = 4096).

In this experiment, it takes on average 10ms to encode a document fragment and 15ms to classify a fragment (for maximum fragment size of 4096 byte) on a regular laptop (core i7, 16GB RAM).

Table IV
MISCLASSIFIED SENSITIVE DOCUMENT FRAGMENTS

Size	*Mormon*	*Dyncorp*	*TM*	*Enron*	*Snowden*
512	5 / 2740	36 / 2540	35 / 671	1117 / 5000	279 / 10250
1024	0 / 2740	4 / 2531	1 / 642	371 / 5000	17 / 9330
2048	0 / 147	0 / 298	0 / 234	153 / 5000	4 / 7307
4096	0 / 137	1 / 241	2 / 129	67 / 5000	3 / 3088

C. Full Document Sensitivity Classification

In this experiment, we evaluate the accuracy for sensitivity classification on full documents in each dataset. In Table V, besides the total number of documents and classification accuracy, we also report the number of false reject, i.e. the number of sensitive documents but wrongly classified as normal ones over the total number of sensitive documents. Our method achieves the accuracy of more than 99.9% for all datasets, except for *Enron* (with 98.356%). Furthermore, the number of false reject document is significantly low for all datasets.

Table V
SENSITIVITY CLASSIFICATION FOR FULL DOCUMENTS

Dataset	Accuracy	False reject	Total documents
Mormon	99.970%	0.0 / 137.0	16681
Dyncorp	99.976%	1.0 / 127.0	16781
TM	99.952%	1.0 / 34.0	16579
Enron	98.356%	33.0 / 5000.0	21473
Snowden	99.900%	2.0 / 514.0	16987

V. CONTENT-BASED DLP SYSTEM

Figure 6 illustrates the overview of our system for data leakage prevention in personal computers in which we deploy the content-based sensitivity evaluation. Apart from malicious threats, sensitive data might be leaked unintentionally in regular users' operations. For examples, a user can save documents with confidential information in a cloud

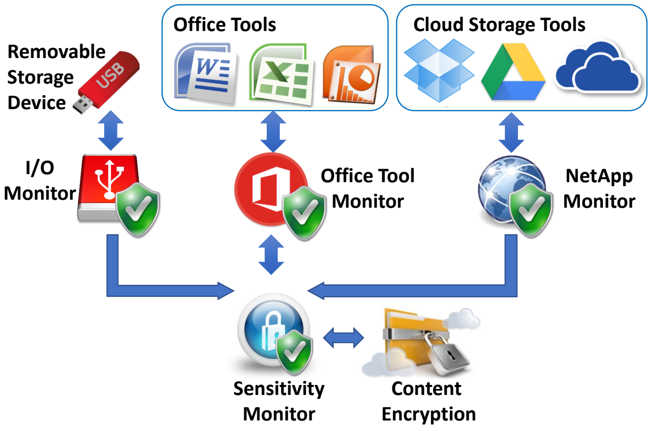

Figure 6. Overview of content-based DLP

storage folder that has been shared with others, or copy such documents into a removable storage device and lose it.

To help users from accidentally leak confidential/privacy data, we develop a system of multiple utilities to monitor users' activities within common applications to check potentially sensitive text fragments. Office Tool Monitor listens to user's actions in Office tools, such as Word, Excel, Powerpoint, to quickly detect a sensitive piece of information when a user is editing a document. We also develop I/O and NetApp Monitors to implicitly follow user's operations with removable storage devices and cloud storage utilities. These three components request Sensitivity Monitor to evaluate the sensitivity level of a short text fragment in operation, raises alarm to users, or encrypt the content upon request.

VI. Conclusion

In this paper, we propose a method to classify the sensitivity of a document for data leakage prevention. Our proposed method aims to better exploit the semantic and content of a document, rather than based on keywords or document format as most of existing methods in DLP. This is an application of our pre-trained document embedding model TD2V[8] into the security and privacy domain. After text fragment vectorization, we use retrieval and query expansion with our Modified Distance [8] to get the list of labeled samples, and determine the sensitivity label (or score) for a new document or text fragment.

We conduct two groups of experiments to evaluate the accuracy of our method to classify document sensitivity in two scenarios: full documents and document fragments. our method can successfully classify full documents in all dataset with the accuracy higher than 99.9% (except for *Enron* with 98.356%). Experimental results also show that our method can correctly recognize a sensitive text fragment with very short length (512 bytes) with the accuracy higher than 90% (except for one dataset *Snowden*). For the fragment length of 4096 bytes, the accuracy is higher than 99% for all dataset (except for Enron with 98.84%). The results demonstrate the potential to use our method to quickly identify a potentially sensitive document just from a short fragment in that document.

We integrate the content-based sensitivity evaluation module in our DLP system to monitor users' activities. Our system monitors operations in regular Office tools, as well as I/O operations with removable storage devices and cloud storage utilities. Currently we continue to optimize the performance of sensitivity evaluation, and implement modules into kernel level to further improve system performance.

Acknowledgment

This research is funded by Department of Science and Technology, Ho Chi Minh city, under grant number 40/2015/HD-SKHCN.

References

[1] S. Alneyadi, E. Sithirasenan, and V. Muthukkumarasamy, "A survey on data leakage prevention systems," *J. Netw. Comput. Appl.*, vol. 62, no. C, pp. 137–152, Feb. 2016. [Online]. Available: http://dx.doi.org/10.1016/j.jnca.2016.01.008

[2] A. Shabtai, Y. Elovici, and L. Rokach, *A Survey of Data Leakage Detection and Prevention Solutions*. Springer Publishing Company, Incorporated, 2012.

[3] InfoWatch Analytical Center, "Global data leakage report, 2016," 2017.

[4] S. Alneyadi, E. Sithirasenan, and V. Muthukkumarasamy, *A Semantics-Aware Classification Approach for Data Leakage Prevention*. Cham: Springer International Publishing, 2014, pp. 413–421.

[5] D. Du, L. Yu, and R. R. Brooks, "Semantic similarity detection for data leak prevention," in *Proceedings of the 10th Annual Cyber and Information Security Research Conference*, ser. CISR '15. New York, NY, USA: ACM, 2015, pp. 4:1–4:6.

[6] J. M. Gomez-Hidalgo, J. M. Martin-Abreu, J. Nieves, I. Santos, F. Brezo, and P. G. Bringas, "Data leak prevention through named entity recognition," in *2010 IEEE Second International Conference on Social Computing*, Aug 2010, pp. 1129–1134.

[7] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents." in *ICML*, ser. JMLR Workshop and Conference Proceedings, vol. 32. JMLR.org, 2014, pp. 1188–1196.

[8] L. Q. Trieu, Q.-H. Tran, and M.-T. Tran, "News classification from social media using twitter-based doc2vec model and automatic query expansion," in *Proceedings of the Eighth International Symposium on Information and Communication Technology*, ser. SoICT '17. ACM, 2017.

[9] M. Mitra, A. Singhal, and C. Buckley, "Improving automatic query expansion," in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '98. New York, NY, USA: ACM, 1998, pp. 206–214.

[10] M. Hart, P. Manadhata, and R. Johnson, *Text Classification for Data Loss Prevention*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 18–37.

[11] Z. Harris, "Distributional structure," *Word*, vol. 10, no. 23, pp. 146–162, 1954.

[12] R. Zhao and K. Mao, "Fuzzy bag-of-words model for document representation," *IEEE Transactions on Fuzzy Systems*, vol. PP, 2017.

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.