Sensitive Data Identification in Structured Data through GenNER Model based on Text Generation and NER

Ji-sung Park
Database Lab.
Hanyang Univ.
Ansan, Republic of Korea
jsdms316@hanyang.ac.kr

Gun-woo Kim
IT Research Institute,
AI Tech Team
LOTTE Data Communication Co.
Seoul, Republic of Korea
gunwoo.kim@lotte.net

Dong-ho Lee†
Database Lab.
Hanyang Univ.
Ansan, Republic of Korea
dhlee72@hanyang.ac.kr

ABSTRACT

A Lot of documents in many organizations from companies to governments are shared on on-premise storage or clouds. And some of those documents may contain sensitive information such as names, social security numbers, addresses and so on. Especially a large amount of sensitive information written in Korean have been leaked nowadays. It can be severe problems to not only individuals but also many organizations. Therefore, for information protection, data loss prevention (DLP) has been needed. DLP systems based on pattern matching were popular in the past. But they have a difficulty handling new type of sensitive data whenever they come. To handle this problem, sensitive data identification with NER is proposed as a useful method of DLP system. By using NER, we can classify the words in a document into categories which consist of name, location and so on. These categories are considered as sensitive information. This approach shows good performance identifying information in unstructured data(e.g. sentences) which have contextual information whereas it has a weakness identifying sensitive information in structured data (e.g. personal names in cells of the table). Actually, a large amount of sensitive information is organized in structured data and the form of structured data varies depending on the document. Furthermore, it also has difficulties identifying data written in Korean because of its characteristics. We proposed a primary preventive measure of DLP by identifying sensitive data in tables of Korean documents combining text generation and NER models regardless of the form of tables and masking them as to share documents without disclosing sensitive information.

†Corresponding Author: dhlee72@hanyang.ac.kr

CNIOT2020, April 24–26, 2020, Sanya, China

ACM ISBN 978-1-4503-7771-3/20/04…$15.00
https://doi.org/10.1145/3398329.3398335

CCS CONCEPTS

• Computing methodologies→Artificial intelligence→Natural language processing→Natural language generation, Information extraction, Phonology/ morphology

• Computing methodologies→Machine learning→Machine learning approaches→Neural networks

KEYWORDS

Sensitive information, Structured Data, Korean language, NLP, named entity recognition, text generation, BiLSTM, CRF, DLP

ACM Reference format:

Ji-sung Park, Gun-woo Kim and Dong-ho Lee. 2020. Sensitive Data Identification in Structured Data through GenNER Model based on Text Generation and NER. In *Proceedings of 2020 International Conference on Computing, Networks and Internet of Things (CNIOT' 20). Sanya, China, 5 pages.* https://doi.org/10.1145/3398329.3398335

1 Introduction

In 2013, there were about 100 million sensitive information leakage including names, social security numbers, phone numbers, addresses, accounts and so on from famous banks or enterprises in Korea. About 64 million sensitive information in Health Insurance Review and Assessment Service(HIRA) which is a quasi government organization of South Korea kept was leaked from 2014 to 2017. It includes names, age, medical and prescription history, etc. As shown above examples, sensitive information leakage problems have emerged as one of the most critical social issues. As most organizations have stored the sensitive information which they collect on premise storage and clouds, there is a high possibility of information leakage if there is no proper countermeasure of information protection.

DLP is a method to prevent data loss. Rule based pattern matching method was popular in the past. But it has low detection rate when new type of data comes. As a solution for this weakness, DLP through Named Entity Recognition(NER) has been proposed by identifying sensitive information. NER classifies data into

categories which are treated as sensitive information. The categories are composed of the name of person, address and so on. Supervised learning based NER system needs to construct a corpus suitable for a new domain through feature extraction from unstructured data like sentences. It requires a lot of time and cost but it shows good performance in classifying words in sentences into appropriate tags which shows whether words are sensitive or not and overcome the weaknesses of rule based pattern matching models. But it still has a problem that it has poor performance in classifying words in the table into tags (e.g. data in MS Excel file). NER models based on Recurrent Neural Network(RNN) predict the tags of word through the sequential information of words in sentence. However, table data doesn't have any sequential information inside itself. It causes low sensitive information detection when using table data directly. Moreover, It is hard to handle data in new types of table with trained data.

A sentence written in English which is an isolated language has not much difference of the meaning after changing of its suffix. In case of English, if the sequence of words is considered well, it shows good performance in Natural Language Processing(NLP) area. Unlike English which is an isolated language, Korean is one of agglutinative languages. Agglutinative languages like Korean make sentences by combining stem and suffix. The changes of suffix in a sentence makes the difference of meaning of a sentence. It makes harder to get good performance in Korean rather than English.

To solve the difficulties of Korean, the methods for deriving the word level embedding vectors, character level embedding vectors, and Part-Of-Speech(POS) tag embedding vectors taking into account the characteristic of Korean have been proposed. Characters and POS tags make systems able to infer the meaning of sentence by grasping the change of words. By adopting this method, we can make GenNER model understand Korean language well.

We propose GenNER model made up of *Text Generation*(*TG*) and *NER* modules. Both of them adopt a model combining Bidirectional Long Short-Term Memory(BiLSTM) and Conditional Random Field(CRF) showing good performance in NLP. As BiLSTM used in NER module is a RNN based model, it also has problems handling with a word in a cell of the table. By generating a sentence from a word intentionally using *TG* module, the input value of *NER* module is changed from word to sentence. And It makes better performance when data organized in the table come. After generating sentences and identifying sensitive data with sentences, our system do masking identified sensitive data to keep them from leakage.

2 Related Work

Rule-based methods are proposed to identify sensitive data in early days of DLP. But these methods require the use of massive dictionary and have low detection precision.

Using rank list which consists of Euclidean distance between text fragment vectors and sensitivity labeled training samples[1] can show how documents are sensitive. The shorter Euclidean distance means that it is more sensitive. But this method does not consider the context of sequence of text fragments in many of documents at all.

Through the recent advances in NLP field, many of studies utilize semantic context information of unstructured data in model-based methods. These model-based methods with dynamic policies can detect sensitive data in a more effective manner rather than rule-based methods.

The model-based methods using NER have been studied recently [2],[3]. The method with Freeling API which is natural language processing tools and consists of Naïve Bayes classification, artificial neural network, support vector machine, decision tree and classification rules shows the applicability of NER model to detect sensitive data [2]. A method with hierarchical levels of granularity can detect not only sensitivity of tokens but also sentences and documents [3]. This method consists of three levels. LSTM-based network is used to detect sensitivity of tokens in sentences in token level. In sentence level, LSTM network and Convolutional Neural Network (CNN) are used. Bag-of-Words, Latent Dirichlet Allocation (LDA) and document embeddings are used to detect sensitivity of documents. Furthermore, the advanced methods using BiLSTM-CNN-CRF model show improved performance for labeling its tags [4],[5]. The method of [5] is specialized in NER for Korean sentences especially. But those NER models are hard to detect sensitive data organized in structured data like the table as each word in cells of the table has no context information. To solve this problem, text generation is needed.

Many methods for text generation have been proposed. Sequence to sequence translation mode l[6] which has Encoder-Decoder structure can be one of them. But the input sentence is too long to fit into the fixed-size vectors. Attention model [7] has been proposed to solve this problem. The input source of these kinds of models is a form of sentence and it requires much more contextual information rather than our system which use words inside a cell of table as an input source. However, BiLSTM model [8] enables to generate sentences with a keyword. This method does not require a bunch of information unlike other methods.

3 Model

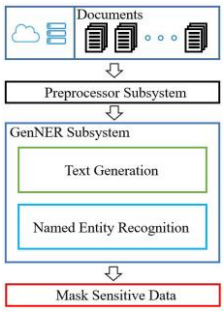

Figure 1: The architecture of sensitive data identification system

Figure 1 shows the overall architecture and process of our sensitive data identification system. Preprocessor makes feature vectors from input words. These feature vectors are used as an input of the first part of GenNER subsystem called *Text Generation* module. Then, the output is the vectors of next words predicted. Through the iteration of *TG* module, the feature vectors that represent a sentence is generated. The sentence feature vectors are used as an input of *NER* module which is the second part of GenNER subsystem. We can get NER tags to identify sensitive data through *NER* module. Finally, masking sensitive data is performed to prevent data leakage.

Preprocessor subsystem is preceded to GenNER subsystem. It tokenizes the word according to its morpheme using Mecab which is a famous Korean morpheme analyzer. It returns tokenized words and POS tags for those words. And then the tokenized words are split into characters. By vectorizing the tokenized words, characters and POS tag through embedding methods like Word2Vec, fastText and glove, embedding vectors of those can be extracted. After that, GenNER subsystem uses the concatenation of tokenized words, characters, POS tags embedding vectors as the input. Figure 2 shows the input feature which consists of embedding vectors of word, its POS tag and characters.

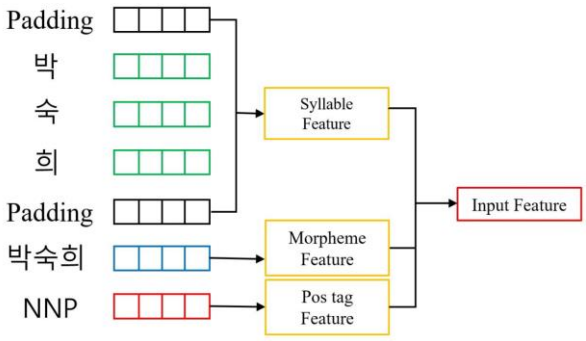

Figure 2: Input feature extraction in Preprocessor

As depicted in Figure 3, both *TG* and *NER* modules in GenNER subsystem have used BiLSTM-CRF method. And, they show quite good performance in both of sentence generation and named entity recognition that will be shown in experiment part.

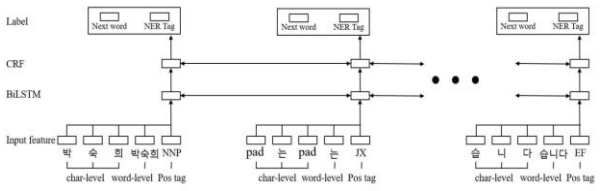

Figure 3: BiLSTM-CRF model of TG and NER modules

BiLSTM consists of two layers which are propagating network called LSTM. BiLSTM uses one layer for forward and the other for backward. These two LSTM networks memorize the information about sentence from both direction. This architecture enables to capture the information previous step and next step respectively and merge the two hidden states to get output. The formulas of LSTM unit at time t when input feature is given x are:

$$f_t = \sigma(W_{xh\_f}x_t + W_{hh\_f}h_{t-1} + b_{h\_f}) \quad (1)$$

$$i_t = \sigma(W_{xh\_i}x_t + W_{hh\_i}h_{t-1} + b_{h\_i}) \quad (2)$$

$$o_t = \sigma(W_{xh\_o}x_t + W_{hh\_o}h_{t-1} + b_{h\_o}) \quad (3)$$

$$g_t = \tanh(W_{xh\_g}x_t + W_{hh\_g}h_{t-1} + b_{h\_g}) \quad (4)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (5)$$

$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

$$y = \mathrm{argmax}_y(y|x) \quad (7)$$

CRF layer uses the output of BiLSTM as input to classify its tag. CRF is based on Bayes rule. Formula (7) is used to find the maximum value of y when x is given. x means data and y means its label. Therefore, CRF is used to find label y the most relevant to x when data x is given.

Both of *TG* and *NER* modules have a common in using BiLSTM-CRF but there are few differences. First, *TG* module predicts the next word to generate the sentence when input word comes whereas *NER* module predicts NER tag of its word to detect whether the word is sensitive or not. Furthermore, *TG* module has generation loop. It iterates whenever next word is generated until end symbol comes.

4 Experiment

4.1 Input data

As the aim of this model is to identify sensitive data and prevent data leakage, the experiment is performed with real public documents on Seoul Information Communication Plaza. This site shares real documents kept in Seoul Metropolitan Government. We collected more than 100 documents including sundry expenses and those are organized in tables. These files are composed of 35,870 word pieces which are processed with Mecab morpheme analyzer. BIO tagging is used for named entity recognition to indicates the end of named entity by adding BIO tags which mean Begin, Inside, Outside of named entities. The files consist of 546 B-PER, 138 I-PER, 337 B-LOC, 874 I-LOC, 689 B-ORG, 1921 I-ORG, 31365 O tags. Using those words with NER tags, 12,000 sentences are generated through rule based sentence generation model. These sentences are tokenized and embedded through Preprocessor and used as the input feature. We use Word2Vec with skip gram as an embedding method. Applying various embedding methods will be a field for future research.

4.2 Text Generation Module (TG Module)

Table I: TG module Performance

TG module	Performance Measure		
	Precision	*Recall*	*F1 score*
Word level	0.83	0.84	0.83
Word level + POS tag	0.85	0.85	0.85
Character level	0.87	0.85	0.86
Character level + POS tag	0.94	0.96	0.95

As an input vector is prepared through Preprocessor subsystem, it goes to the first step of GenNER subsystem, *TG* module.

The input feature of *TG* module represents each words in a sentence and its label data is vectors of next word in sentence. Using BiLSTM-CRF, when the word vector is given, the predicted output is the vectors of a next word. Finally, generation loop iterates its sequence getting next words until a sentence is finished.

Table I shows the performance depending on the configuration of input vectors. Whether POS tag embedding vector are added or not shows slight differences in performance. It shows better performance when using POS tags. Furthermore, by adding character level embedding vectors for deriving word level vectors, we can get the best performance in *TG* module.

Table II: Examples of TG Module

Word	Sentence	
박지성	Kor	박지성은 상기사항을 확인하였습니다.
Jisung Park	Eng	Jisung Park confirmed the context of this document.
서울	Kor	서울특별시 종로구 창신동에 위치하고 있습니다.
Seoul	Eng	It is located in Changsin-dong Jonglo-gu Seoul.

Table II shows some results of *TG* module from the words '박지성' which is the name of a person and '서울' that is the capital city of South Korea. *TG* module intends to be trained to generate a sentence in a form of "[PER]은 상기사항을 확인하였습니다." which is translated into "[PER] confirmed the context of this document". In case of address, it is trained in the same way with name to generate "[LOC]에 위치하고 있습니다." whose meaning is "It is located in [LOC]". The sentences generated are used as input of *NER* module to recognize if it is sensitive.

4.3 Named Entity Recognition Module (NER Module)

Table III: Named Entity Recognition module (NER module)

NER module	Performance Measure		
	Precision	*Recall*	*F1 score*
Word level	0.78	0.76	0.77
Word level + POS tag	0.79	0.78	0.78
Character level	0.94	0.95	0.94
Character level + POS tag	0.96	0.96	0.96

TG module unlike *NER* module uses information of sentences only. But *NER* module has no idea which tags are represented from the input word. Thus, NER tags for each word in a sentence should be prepared in advance manually. In training process, it shares the input feature with *TG* module but it also uses the output of *TG* module as an input feature in test process.

Table III shows the performance of *NER* module depending on configuration of input vectors. By using POS tag embedding vectors, we can get slightly better performance in *NER* module. But when deriving word level embedding vectors from character level embedding vectors, we can get better performance comparing to the case which doesn't use character level embedding vectors. The best performance is shown when using both character level and POS tag embedding vectors are used.

Table IV: Example of NER Module

Sentence	서울특별시 종로구 창신동에 위치하고 있습니다.	
	It is located in Changsin-dong, Jonglo-gu, Seoul	
Words	*Real*	*Prediction*
서울특별시 (Seoul)	B-LOC	B-LOC
종로구 (Jonglo-gu)	I-LOC	I-LOC
창신동 (Changsing-dong)	I-LOC	I-LOC
에	O	O
위치	O	O
하	O	O
고	O	O
있	O	O
습니다	O	O
.	O	O

Table IV shows an example of *NER* module with using a sample sentence generated by *TG* module. The words with tags like LOC, PER and so on are considered as sensitive data which need to be protected from leakage.

4.4 Result

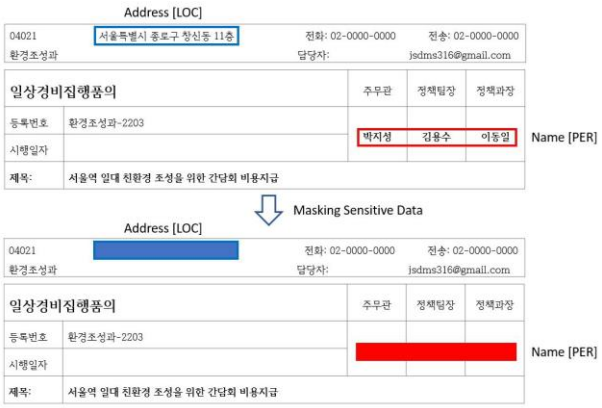

Figure 4: Sensitive Data Identification

Figure 4 shows a result of our system when a document with structured data comes. The input document used in Figure 4 is a fake document for not showing sensitive information of individuals which should not be open to public. But we used real data for the test. By using NER tags from *NER* module, we can figure sensitive data out and mask them.

Table V: Performance Result

	Performance Measure		
	Precision	***Recall***	***F1 score***
named entity recognition model with row data	0.75	0.74	0.74
GenNER model	0.90	0.92	0.91

Table V shows the performance comparison the result of GenNER model and named entity recognition model using BiLSTM-CRF without text generation which uses row data parsed from tables in documents as an input directly. Both of them used word level embedding vectors with character level and POS tag embedding vectors which shows good performance with *TG* and *NER* modules. Comparing to named entity recognition model with row data, it shows a remarkable performance improvement through GenNER subsystem.

5 Conclusion

We have proposed a method called GenNER whose aim is to detect sensitive information included in structured data like personal names and address in cells of a table as a method of DLP. The main idea of GenNER model is to recognize named entities in structured data which have no contextual information by generating a sentence using a word in structured data like a column name of the table. GenNER model combines the text generation process which generates sentences with the word in structured data and the process for named entity recognition which predicts categories of words within generated sentences. By getting categories classified as sensitive information like PER, LOC, ORG and so on, we can identify sensitive information in structured data and mask them to prevent data leakage. However, it still has problems such as corpus construction, Out Of Vocabulary (OOV) which are chronic problems when using supervised learning methods. We will solve these problems and improve the performance of GenNER subsystems by applying other state-of-the-art embedding methods and deep learning models to keep sensitive information from leakage.

ACKNOWLEDGMENTS

"This research was supported by the MISP (Ministry of Science, ICT & Future Planning), Korea, under the National Program for Excellence in SW)(2018-0-00192) supervised by the IITP(Institute for Information & communications Technology Promotion)"(2018-0-00192)

REFERENCES

[1] Trieu, Lap Q., et al. "Document sensitivity classification for data leakage prevention with twitter-based document embedding and query expansion." 2017 13th International Conference on Computational Intelligence and Security (CIS). IEEE, 2017

[2] Gomez-Hidalgo, Jose Maria, et al. "Data leak prevention through named entity recognition." 2010 IEEE Second International Conference on Social Computing. IEEE, 2010.

[3] Ong, Yuya Jeremy, et al. "Context-aware data loss prevention for cloud storage services." 2017 IEEE 10th International Conference on Cloud Computing (CLOUD). IEEE, 2017

[4] Ma, Xuezhe, and Eduard Hovy. "End-to-end sequence labeling via bi-directional lstm-cnns-crf." arXiv preprint

[5] Kim, GyeongMin, et al. "Constructing for Korean Traditional culture Corpus and Development of Named Entity Recognition Model using Bi-LSTM-CNN-CRFs." Journal of the Korea Convergence Society 9.12 (2018): 47-52Conference Name:ACM Woodstock conference

[6] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems. 2014.

[7] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

[8] Song, Ziyao, et al. "A Neural Network Model for Chinese Sentence Generation with Key Word." 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC). IEEE, 2019.