Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

CASSED: Context-based Approach for Structured Sensitive Data Detection

Vjeko Kužina [a], Ana-Marija Petric [b], Marko Barišić [a], Alan Jović [a],[*]

[a] University of Zagreb, Faculty of Electrical Engineering and Computing, Unska 3, 10000, Zagreb, Croatia
[b] Legit, Pavleticeva 1, 10000, Zagreb, Croatia

ARTICLE INFO

Keywords:
Sensitive data detection
Privacy protection
Structured data
Machine learning
Transformers
Context-based detection

ABSTRACT

The need for sensitive data detection and identification has increased in recent years. Sensitive data detection and identification are necessary steps for privacy protection. The focus in this field has been on unstructured data detection using natural language processing (NLP) approaches, while there has been little progress in the field of structured data. Most of the structured data approaches consider independent feature representations of cells, without taking potentially relevant context into account. In this work, we introduce a novel context-based approach named **CASSED**, which stands for **C**ontext-based **A**pproach for **S**tructured **SE**nsitive **D**ata Detection. CASSED addresses the problem of sensitive data detection in structured data through the lens of NLP, using the transformer-based BERT method. Our approach aims to actively capture relations both within and between cells in the same column as the assumption is that the data present in the same column in a table are mostly very similar. CASSED works as a classifier for columns in database tables with the task of predicting a label or multiple labels for different types of sensitive data that a column may represent. Since there is no officially recognized dataset for the task, we compared CASSED on datasets used for similar tasks from related work. Furthermore, we created our own dataset focused on sensitive data to evaluate CASSED. Our method outperformed methods from related work both on their datasets and achieved significantly better results on our own dataset compared to our baseline model as well as models from related work. Our research suggests that treating structured data as context-rich is a viable strategy for sensitive data detection and identification.

1. Introduction

The protection of personal data features prominently both in academic and business circles. The General Data Protection Regulation within EU (GDPR, 2016) sparked not only the widespread debate on data privacy and protection but also the creation of many different privacy acts and regulations worldwide, such as the California Consumer Privacy Act (CCPA, 2018) in California, or the General Personal Data Protection Law (LGPD, 2018) in Brasil. All of these regulations aim to give individuals a relatively high degree of control over their personal data and impose strict guidelines on the data collectors. As a consequence of the application of these regulations, the need for sensitive data detection and privacy protection has increased (Spector, Norvig, Wiggins, & Wing, 2022). Privacy protection is challenging in many ways, as companies collect vast amounts of data often without knowing exactly what is being collected or how to effectively search for or retrieve personal data. While some of the collected data are in the form of unstructured text, the majority are in the form of structured databases which contain multiple tables consisting of cells organized into rows and columns.

This paper addresses the challenges of detecting sensitive data in structured databases. There is a need to automate this process because the amount of data stored in structured databases is considerable and the regulations about what is considered sensitive data are always changing (Quinn & Malgieri, 2021). Specifically, the problem of structured sensitive data detection is going through the columns in a database table and determining for each of them whether they contain sensitive data or not, as well as what type(s) of sensitive data they are. The formulation of the problem can be viewed as a subtype of the semantic column labeling problem (Trabelsi, Cao, & Heflin, 2021), which is a more general problem with strong similarities to the named entity recognition problem (Marrero, Urbano, Sánchez-Cuadrado, Morato, & Gómez-Berbís, 2013), since named entities must be detected in the data.

For any solution tackling sensitive data detection to be successful, it is imperative to correctly interpret the meaning of a cell, as well as take the context from surrounding cells into account. In recent years, different attempts have been made to solve this problem. The initial approaches relied heavily on rule-based heuristics such as capital

[*] Corresponding author.
E-mail addresses: vjeko.kuzina@fer.hr (V. Kužina), ana-marija.petric@inteligencija.com (A.-M. Petric), marko.barisic@fer.hr (M. Barišić), alan.jovic@fer.hr (A. Jović).

https://doi.org/10.1016/j.eswa.2023.119924
Received 22 July 2022; Received in revised form 7 February 2023; Accepted 17 March 2023
Available online 21 March 2023

letters, regular expressions (regex), or lists of predefined entities (via lookup tables). While rule-based methods can correctly identify some of the sensitive data types, they become challenging to use in the detection of certain data types (e.g., physical addresses or phone numbers) which have a large number of different formats (Ye, Chen, Wang, Dillig, & Durrett, 2020). Furthermore, rule-based approaches inherently lack context and understanding. Different machine learning methods have been developed to address the latter problem. Most of these approaches attempt to circumvent the aforementioned issues of context and understanding for tabular data by either applying machine learning methods directly to the cells, relying on column statistics or character distributions, or considering other columns in the database to aid in the classification. While these methods do work to some degree, in a recent work (Wu, Wu, Qi, & Huang, 2020), it has been shown that the solutions to the structured sensitive data detection could benefit from using Natural Language Processing (NLP) and leveraging the context inside of and between cells in the table to deduce embeddings of each cell. Although some of the related approaches like SeLaB (Trabelsi et al., 2021) and TaBERT (Yin, Neubig, Yih, & Riedel, 2020) do use NLP and context by averaging cell embeddings over columns or generating embeddings from the whole row, they do not allow the NLP models to capture multiple cells from the same column at the same time, and thereby directly use tabular context to deduce the appropriate label of the whole column.

To solve the structured sensitive data detection problem and its automation, we propose a novel method named **CASSED** (**C**ontext-based **A**pproach for **S**tructured **SE**nistive **D**ata Detection), which creates column context by combining column metadata and cell values and integrates them into a single input for a natural language embedding model (BERT). Subsequently, BERT (Devlin, Chang, Lee, & Toutanova, 2018) allows the classification of individual columns into one or more labels in a way that the natural language embedding can consider multiple cells of the same column simultaneously. CASSED also employs rule-based methods to aid in the classification of formulaic data types such as social security numbers or credit card numbers.

The contribution of this work is the following:

- A novel method for structured sensitive data detection is proposed that classifies columns into one or more labels using an active context-based approach in addition to traditional rule-based heuristics;
- A dataset is created for handling the problem that can be used as a standard benchmark for the related methods due to a lack of such a dataset for structured sensitive data detection;
- The proposed method's classification model demonstrates clear superiority when compared with a baseline model and with models from the related work on the novel dataset;
- The proposed method's classification model outperforms models from related work on their own published datasets that are used to handle the more general problem of semantic column labeling.

The remainder of this paper is structured as follows. Section 2 provides a brief overview of various approaches to the structured data interpretation and labeling problem, with specific focus on the detection of sensitive data. In Section 3, we first explain the dataset which we created and used for training our model, as well as provide further details about the model itself. Section 4 details the experiments we conducted in order to correctly identify sensitive data. In Section 5, we discuss the challenges encountered and the results obtained. Finally, Section 6 concludes the paper.

2. Related work

As discussed, the main focus of our work is the efficient detection of sensitive data in structured datasets. However, research and data for this specific problem are not widely available. Therefore, we expanded our search and included research into the related semantic column labeling tasks as well as the creation of embeddings for structured data. Namely, the interpretation and classification of structured datasets can be achieved in a variety of ways, depending on both the task and the approaches used to solve the task. The approaches include rule-based and machine-learning methods which analyze the data stored in the structured datasets and aid in solving the considered task.

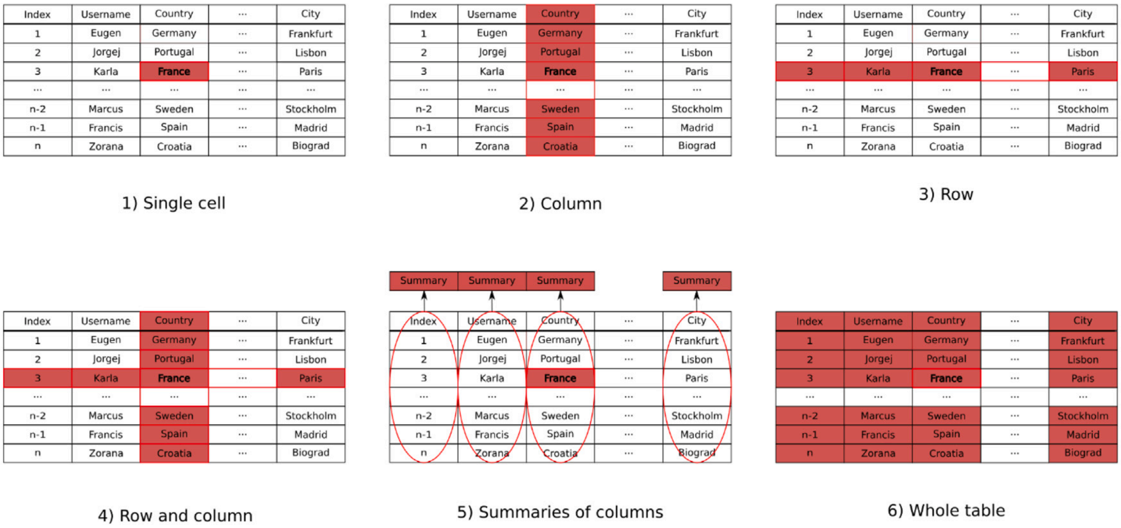
Besides the basic differentiation between using rule-based and machine-learning approaches for detection problems in structured datasets, there are several other ways to differentiate approaches (Vušak, Kužina, & Jović, 2021). The first separation looks into the way machine learning models use the values inside the cells, which is also called intra-cellular context. In this sense, some machine-learning approaches such as character or word embeddings look at characters or words separately and encode them without considering other characters or words in the cell, and later join or average them to create the final embeddings. Other machine learning approaches, such as BERT, use contextualized word embeddings and consider all words inside the cell simultaneously, and pay attention to what is important while looking at the whole input. The second way of differentiation of approaches looks at the way the approaches use inter-cellular context when identifying sensitive information contained within a cell. Inter-cellular context refers to the use of other cells inside the table to create the embeddings of cells. The most often used inter-cellular context approaches are shown in Fig. 1. In theory, the detection results would be optimal if the models employ the approach shown under (6) in Fig. 1, namely taking the whole table as context and using all the available information for the detection of sensitive data types in each cell. However, for practical purposes, this approach is not computationally feasible for now due to its high time and space requirements, especially for large database tables. Thus, the approaches from related work implement different ways of taking context into account, depending on what is considered to be the most useful or computationally the least demanding for their task. The third differentiation is in the way inter-cellular context is used, namely either passively or actively, where active context refers to the creation of embeddings with insight into multiple cells directly, and the passive context approach which generates cell embeddings separately and later joins or averages them.

In Table 1, we provide an overview of the related work along with the methods used and limitations observed.

Some commercial approaches like Trifacta (Trifacta, 2014) and Power BI (Microsoft, 2016) mostly use rule-based methods, while other approaches such as Cloud DLP (Google, 2018) and PII Catcher (PII, 2018) use both rule-based and machine learning methods on database cells. These approaches fall under the simplest category (1) in Fig. 1, with none of them taking any inter-cellular context into account.

SIMON (Azunre et al., 2019) attempts to solve the semantic column labeling problem by employing a character-level convolutional neural network (CNN) together with a long short-term memory (LSTM) network (Hochreiter & Schmidhuber, 1997) to produce embeddings of individual cells, combine them into a singular embedding and subsequently classify the column into one of the possible labels. SIMON, however, does not use any kind of context and only averages the cell values of each column to predict the final label.

Sherlock (Hulsebos et al., 2019) aims to consider relations between cells and formulates more complex concepts of context. To this end, Sherlock uses a deep neural network architecture that does not only use the current cell in the table to generate its features but also incorporates context by considering all other cells in the same column through a modified version of Paragraph Vectors (Le & Mikolov, 2014) which are generated beforehand. This approach is shown under (2) in Fig. 1, and includes taking into account only the summary of the current cell's column. Together with the paragraph vectors, Sherlock also employs statistical features of the current column such as character distributions and average cell lengths.



Index	Username	Country	···	City
1	Eugen	Germany	···	Frankfurt
2	Jorgej	Portugal	···	Lisbon
3	Karla	France	···	Paris
···	···	···	···	···
n-2	Marcus	Sweden	···	Stockholm
n-1	Francis	Spain	···	Madrid
n	Zorana	Croatia	···	Biograd

1) Single cell

Index	Username	Country	···	City
1	Eugen	Germany	···	Frankfurt
2	Jorgej	Portugal	···	Lisbon
3	Karla	France	···	Paris
···	···	···	···	···
n-2	Marcus	Sweden	···	Stockholm
n-1	Francis	Spain	···	Madrid
n	Zorana	Croatia	···	Biograd

2) Column

Index	Username	Country	···	City
1	Eugen	Germany	···	Frankfurt
2	Jorgej	Portugal	···	Lisbon
3	Karla	France	···	Paris
···	···	···	···	···
n-2	Marcus	Sweden	···	Stockholm
n-1	Francis	Spain	···	Madrid
n	Zorana	Croatia	···	Biograd

3) Row

Index	Username	Country	···	City
1	Eugen	Germany	···	Frankfurt
2	Jorgej	Portugal	···	Lisbon
3	Karla	France	···	Paris
···	···	···	···	···
n-2	Marcus	Sweden	···	Stockholm
n-1	Francis	Spain	···	Madrid
n	Zorana	Croatia	···	Biograd

4) Row and column

Summary	Summary	Summary		Summary
Index	Username	Country	···	City
1	Eugen	Germany	···	Frankfurt
2	Jorgej	Portugal	···	Lisbon
3	Karla	France	···	Paris
···	···	···	···	···
n-2	Marcus	Sweden	···	Stockholm
n-1	Francis	Spain	···	Madrid
n	Zorana	Croatia	···	Biograd

5) Summaries of columns

Index	Username	Country	···	City
1	Eugen	Germany	···	Frankfurt
2	Jorgej	Portugal	···	Lisbon
3	Karla	France	···	Paris
···	···	···	···	···
n-2	Marcus	Sweden	···	Stockholm
n-1	Francis	Spain	···	Madrid
n	Zorana	Croatia	···	Biograd

6) Whole table

Fig. 1. Approaches to inter-cellular context representation. In all cases, the cell with the value 'France' is the currently looked-at cell. (1) Only the current cell is taken into account, there is no context. (2) The current cell and the column of that cell are taken as context. (3) The current cell and the row of the cell are taken as context. (4) The current cell, the row, and the column are taken as context. (5) Summaries for all columns are created a priori and they are given as context together with the current cell. (6) All the cells in the table are given as context.

Table 1
The description of related work with the methods they are using, and what in our mind are the limiting factors or shortcomings.

Description of related work approaches and limitations		
Related work	Methods	Limitations
Trifacta (Trifacta, 2014), Power-BI (Microsoft, 2016)	Rule-based methods	No natural language understanding No context No adaptability
Cloud DLP (Google, 2018), PII catcher (PII, 2018)	Rule-based methods Basic machine learning methods on singular cells	No intra-cellular context No inter-cellular context
SIMON (Azunre et al., 2019)	Character level embeddings of individual cells	No intra-cellular context No inter-cellular context
Sherlock (Hulsebos et al., 2019)	Single cell embeddings Static column embeddings Statistical column features	No intra-cellular context No active inter-cellular context
SATO (Zhang et al., 2019)	Deep neural networks Static context of other columns and statistics	No intra-cellular context understanding No inter-cellular context
SeLaB (Trabelsi et al., 2021)	BERT for intra-cellular context Static inter-cellular column context	No active inter-cellular context
TaBERT (Yin et al., 2020)	BERT for intra-cellular context Active inter-cellular context of entire row	Created for question answering task No context of other values in column
TABBIE (Iida, Thai, Manjunatha, & Iyyer, 2021)	BERT for intra-cellular context Static inter-cellular context from row and column	Created for corrupt cell detection Not looking at multiple cells while creating embedding with natural language

SATO (Zhang et al., 2019) extends on Sherlock (Hulsebos et al., 2019) by incorporating a Topic Prediction model which works on the entire table. In such a way, SATO also uses other columns in the database to extend the context of a cell, which can be seen under (5) in Fig. 1. While both SATO and Sherlock consider table context, they do so in a passive way, meaning that the context is generated beforehand and the model is not allowed to actively look at other cell values while creating an embedding for the current cell.

SeLaB (Trabelsi et al., 2021) approaches the semantic column labeling problem using a two-step processing. In the first step, the method uses BERT to generate embeddings and classify each cell individually, and subsequently calculating the most likely label from these. In the second step, SeLaB repeats the first step with additional information of other predicted column labels from the first step, thereby using the context of all other columns to determine the label of each column. This approach can be seen as a variation of the approach under (5) in Fig. 1.

TaBERT (Yin et al., 2020) tackles a slightly different problem as it aims to answer the question in the form of a sentence and predict which cells from the table give the best answer to the question. Its value for the current problem is in giving a glimpse into a possible way to incorporate dynamic context. TaBERT attempts to solve the question-answering problem by calculating the similarity of the sentence to a row in the table and then feeding both the sentence and the most similar row into BERT (Devlin et al., 2018), as shown under (3) in Fig. 1, thus allowing the model to consider all cells inside of the same row at the same time, effectively allowing the model to learn what it needs to pay attention to in its context.

TABBIE (Iida et al., 2021) is a recent method that first creates an embedding for each cell separately with BERT. Afterwards, individual embeddings of cells in the same row are taken as input into the transformer. The process is repeated for each individual cell in the column. In the end, the outputs are averaged to obtain the embedding which takes into account the context of both current row and column. This is a strategy depicted under (4) in Fig. 1.

Our idea to tackle the problem of structured sensitive data detection through the lens of NLP was inspired in part by TABBIE's idea of using whole columns as context for BERT and by TaBERT's idea to concatenate cells as input and give them straight to BERT in their tokenized format, without creating embeddings for individual cells first, which is an approach depicted in Fig. 1 under (2). This resulted in the removal of an intermediary step of single-cell embedding creation. By removing that step, the features of the input are entirely preserved for BERT and its contextual embedding method. This is an active inclusion of context, and is the main contribution of our proposed method with respect to related work. Our approach concatenates the cell values of a whole column and tokenizes them to create an input for BERT. Afterward, BERT produces embeddings for each token of the input and later on deduces the final classification of the column.

3. Materials and methods

The datasets created and used by aforementioned methods tackle a very broad problem of semantic column labeling. While CASSED can solve tasks of this nature, the idea behind creating it is to tackle the specific challenge of sensitive data detection in structured datasets for which there is no widely available dataset. To confront the lack of a relevant dataset, we opted to create a hybrid dataset, consisting of both synthetic (generated) data and pseudo-anonymized real-world data, which incorporates a wide array of sensitive data types. In the continuation of this section, we will first describe the data challenges we faced, then proceed with the description of our created dataset, and finally elaborate on the CASSED method.

3.1. Data

3.1.1. Challenges in sensitive data detection datasets

Building a machine learning model to recognize sensitive data relies heavily on the availability of real-world datasets containing personal information. However, these datasets are, by definition, not readily available to the public. Furthermore, the use of real-world datasets in training and testing machine learning models can lead to potential security issues, such as the extraction of sensitive data from the classifier itself (Ateniese et al., 2013) or the possibility of extracting the information from the neural networks through statistical inference, as described in Dwork, Smith, Steinke, and Ullman (2017). These considerations necessitate the creation of synthetic datasets. Synthetic datasets are commonly used in disciplines where privacy is a concern, such as medicine (Chen, Lu, Chen, Williamson, & Mahmood, 2021) and computer vision (Nikolenko, 2019). Synthetic data can be used both to train the model and to augment a real-world dataset. In related work, synthetic datasets were created by either a direct use of tools such as Faker (Faker, 2021) or by taking columns from other datasets such as VizNet (Hu et al., 2019) and refining them to take a subset of columns matched to DBpedia column types. While these datasets can be decent as a comparison for machine learning models in general, they do not represent sensitive data types, and therefore are not the best possible representation of data to handle our problem.

Furthermore, the datasets from related work do not seem to incorporate column headers for various reasons. Most of the related work, such as Sherlock (Hulsebos et al., 2019) or SIMON (Azunre et al., 2019), either state that column headers are not reliable sources of information or they use column header data to generate the column labels. While we acknowledge that column headers can sometimes be empty, provide useless or even misguiding information and that models should not rely solely on them, we strongly disagree with the notion that they are useless in most cases. Instead, we argue that column headers very often carry substantial information about the column data type. Additionally, column headers are almost always present in real-world data in some form and should be incorporated into synthetic datasets and used by models for context creation.

We also draw attention to the problem of VizNet (Hu et al., 2019) and Faker (Faker, 2021) datasets, which have only one label for each column. This is different from real-world data sets, where it is often quite possible for one column to contain multiple types of sensitive data labels.

Lastly, there are multiple other challenges in real-world structured data, which can potentially limit the efficiencies of machine learning models and require careful data preparation. Some of these challenges include input errors, inconsistent column headers, human errors, the use of shorthand or business-specific labels for columns, etc.

3.1.2. DeSSI dataset description

To counter these obstacles and to facilitate the use of CASSED in relational databases for sensitive data detection, we created our own relational data models to train and test the approach. These models aim to simulate the real-world environment. The aforementioned structured data challenges of missing or misleading column headers were simulated in our dataset so that the model would not have to rely heavily on column headers. The constructed dataset consists of snippets of personal data aggregated from various open-source datasets (e.g., from Kaggle), synthetic data generated by Python packages such as Faker (Faker, 2021), and pseudo-anonymized real-world data provided to us by an organization under a strict third-party confidentiality agreement.

The sensitive data types in question can be seen in Table 2 with all the remaining information, which is not sensitive, being labeled as 'Other data'. While the dataset incorporates column headers, we made sure that they are sometimes randomized strings which give no information, or even give misleading information, to account for possible bad practices or errors observed in real-world datasets. The dataset consists of over 31,000 database columns with 100 rows, together with column headers. The data are presented in the format of comma-separated values and published on Kaggle (DeSSI, 2022). The dataset, which we named DeSSI (**D**ata**s**et for **S**tructured **S**ensitive **I**nformation), is randomly split in ratios of 60/20/20 percent among training/validation/test datasets. The labels in our dataset are manually labeled columns with either a single label if the column contains one type of sensitive data or multiple labels to cover the cases where the column contains multiple types of sensitive data.

Since the goal of our model is to detect sensitive types of information, the amount of possible labels in the dataset is restricted to sensitive data types and is thus smaller than the number of labels in other datasets, which in most cases contain general semantic types and not sensitive data. Since the number of labels in our dataset is smaller and also mostly easier to detect than general purpose labels present in datasets of related work, this also leads to higher results on our dataset than on the datasets from most of the related work.

Table 2
Types of sensitive data considered.

Class	Description
Other data	Everything that does not fall under any sensitive data type
Phone number	Various supported formats of landline and mobile phone numbers
Address	Multiple formats, can contain the street name, street number, and postal code
Person	Name, Surname, can contain multiple of each or a combination
Email	Email address
NIN	National Identification Number
Date	Various date formats
Organization	Various types, supports extensions, i.e LLC, DD…
GPE	Geopolitical entities such as states, cities, countries, etc.
Geolocation	Longitude and Latitude
SWIFT/BIC	Business identification codes for both financial and non-financial institutions
IBAN	International bank account number
Passport	Passport numbers
Religion	Religions and members of such organizations
CCN	Credit card numbers with all their supported formats
ID Card	Numbers of identification cards
Sexuality	Various types of sexualities
Gender	Various types of genders
Nationality	Nationalities based on countries
Race	Various descriptions of races

To the best of our knowledge, DeSSI is the first widely available dataset for sensitive data detection in structured data. Other available datasets are general as they were created to solve a broader problem, while DeSSI is a more specific dataset focusing on sensitive data. We acknowledge that DeSSI can certainly be improved and expanded upon in the future. However, we do encourage the use of it as a standard dataset for comparison on the problem of sensitive data detection in structured data sources.

3.2. Initialization and input

CASSED attempts to account for the nuances of natural language which often occur within cells, as table cells may contain multiple words, or in some cases, even multiple sentences. To this end, CASSED uses BERT — the well-known language representation model which has extensive capabilities for detection of relations between words. When testing the other existing models mentioned in the related work section, we could not obtain satisfactory results on the tabular data we generated. Therefore, we decided to shift our focus to the approach that worked well on our unstructured data, namely treating the column as a quasi-natural sentence and using context-sensitive transformers. The driving idea behind this approach was the assumption that cells in a column should be similar to each other in some way, and that allowing the model to look at all cell values in a column at the same time might give more information than looking at each cell individually and then averaging. In a large portion of cases, we can also rely on the column header as an indication of what purpose these cells should have. However, we decided against considering the column header as the most important feature as column header naming conventions can include shorthands, or even be missing or misleading in real-world enterprise databases.

To take advantage of these assumptions, our model constructs the input to BERT from the column header together with multiple cell values from the same column, separated by delimiters, as shown in Fig. 2. In such a way, BERT can take multiple cell values at the same time into account, thereby jointly incorporating both the context inside of a single cell, as well as the context between cell values. During the research, we also tested multiple types of delimiters and found that the choice of the delimiter does not significantly affect the performance of our method. While any delimiter works well, we have found an increase in performance when using different delimiters to separate column headers and cell values. This could be expected since they are used for different purposes and give additional information to the model.

3.3. Concatenating columns

Another problem faced by the method is the limited number of tokens that can be entered into BERT. In the default version of BERT, the number of tokens in one sequence is limited to 512. The maximum token count is exceeded in such cases where database cells contain multiple sentences or when tables contain a large number of rows. This problem can be tackled in different ways. One option is to simply truncate the input to the maximum token length. However, this could lead to a loss of valuable information, especially when the values of cells are long and contain dozens of tokens, as this would make the input consist of only a few cell values. The additional unfavorable outcome is that multiple labels in the column are not detected due to the truncated input, as some of them might simply not make the cut. To address these issues, we opt to not truncate, but instead split the information from the whole column into multiple inputs if the token count exceeds the maximum token count, after which each column part is sent to BERT separately and in the end the BERT's output is averaged.

3.4. Processing

After the embeddings are created and the input is finalized, BERT is employed on a batch of column embeddings. BERT's decoder produces a non-normalized prediction (logit), for each label, which is afterwards averaged over all column parts that the column was separated into. After the averaged logits for the whole column are calculated, a sigmoid function is applied to each of the logits to produce normalized probabilities for each class individually. A sigmoid function is used, rather than a softmax function as the task is to find multiple labels if they are present. If a softmax function were to be used, the majority presence of one label might diminish the presence of another label, while a sigmoid function considers all labels separately from each other.

After the probabilities are calculated for each label, they need to exceed the threshold value required for the classification, which, after extensive testing, we have found to be around 0.4. The depiction of the whole CASSED method is shown in Fig. 3.

3.5. Post-processing

As previously mentioned, CASSED aims to detect cases of sensitive data using mainly machine learning and NLP, but in addition, it also uses rule-based methods. These methods often help with the detection of sensitive data types which have a specific and strict form that is

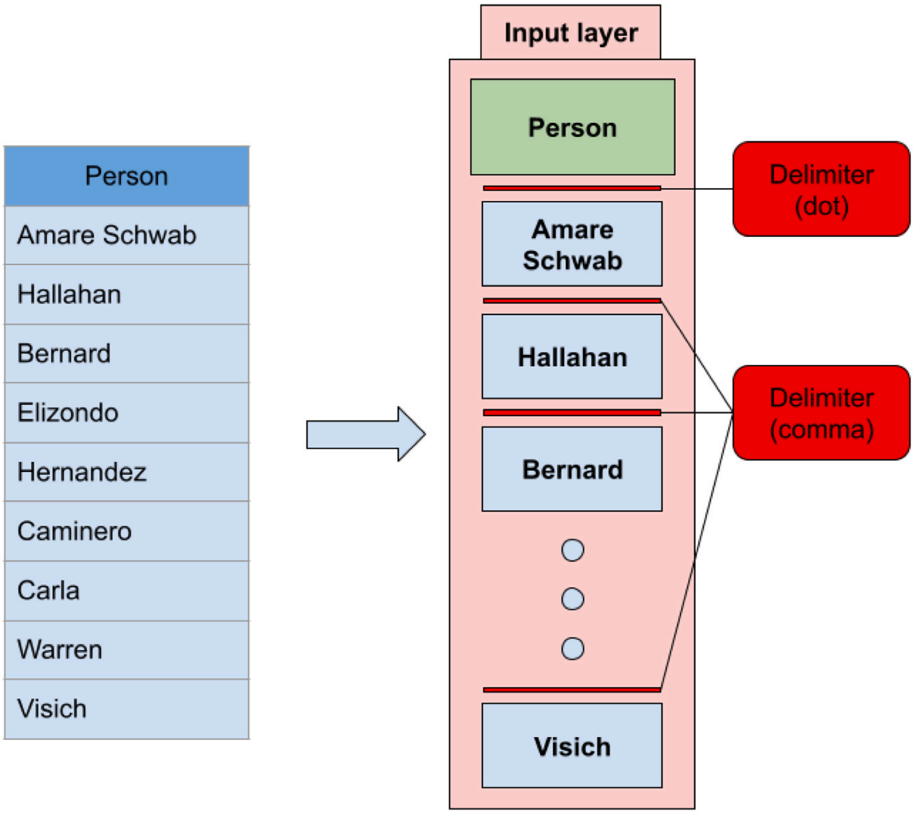


Fig. 2. An example of a column turned into an input for the model. The values of cells are separated by a comma, while the name of the column is at the front, separated by a dot.

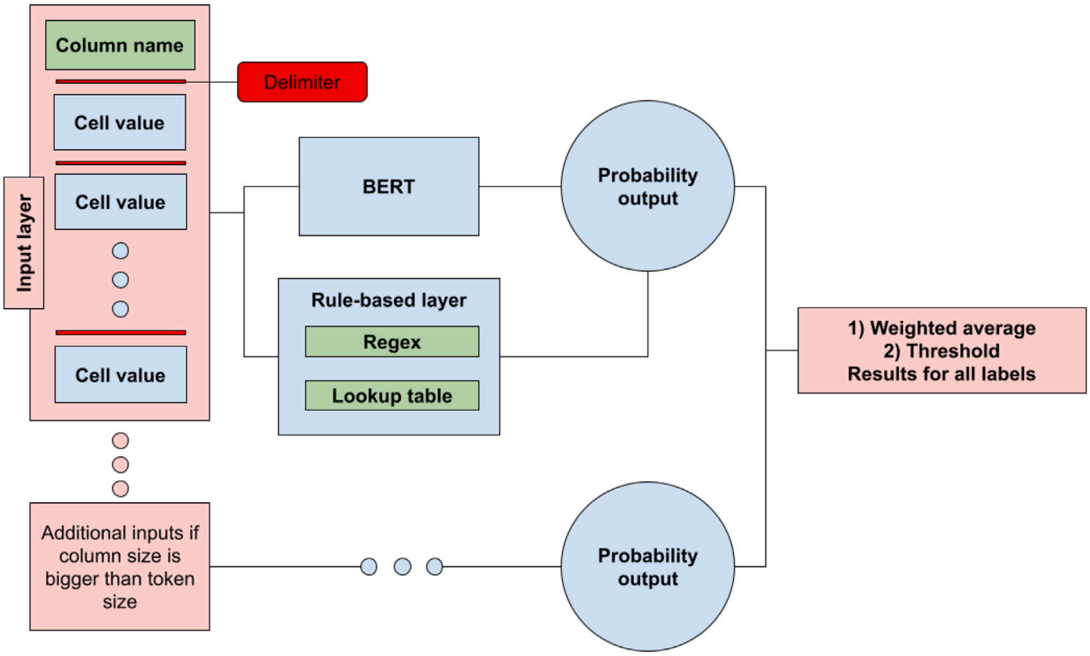


Fig. 3. CASSED method overview — the column is represented as an input and forwarded to both BERT and a rule-based layer separately, after which the probabilities for each sensitive data type are generated and averaged over all the column parts. If the averages exceed a certain threshold, then the column is classified as the according type, allowing multiple labels.

known beforehand, or when sensitive data type only occurs in a subset of values, most of which are known. While machine learning methods are efficient for general sensitive data type detection purposes, they usually struggle with very specific cases, such as deciding if a 16-digit cell value is a social security, credit card, telephone, or some other kind of number. The differences among these sensitive data types are subtle and it can be very hard even for a human to discern them. In these situations, where sensitive data come with very specific limitations and formats, without a lot of context, rule-based methods help significantly to disambiguate them.

Therefore, for sensitive data types such as emails, social security numbers, or credit card numbers, CASSED uses regular expressions, because these types can only occur in specific established formats, and instances of false positives occur in negligible amounts, if at all.

For the values which belong to a subset of possible values, the model uses lookup tables. Such classes are gender, nationality, religion, ethnicity, and so on. There is always a small, finite set of options and all of the values can be listed so that they can always be detected.

In addition to the main flow of classification through BERT, a side flow is created where each input column is passed through rule-based methods that produce a probability for the column to be of that label by looking at the ratio of the number of detected labels inside of the column and the total amount of cells in the column. The generated ratio is treated as a probability of the whole column representing that label type. After the main flow of the model classifies the column into one or multiple classes, the side flow then compares the probabilities it generated and also compares them to the threshold, altering the classification if it exceeded the threshold.

3.6. Baseline

Before creating CASSED, we developed a model using more traditional methods for structured data. This model did not provide satisfactory results, which led us to look for and experiment with more unconventional methods. To compare the results from CASSED, we used our baseline model alongside methods from related work. The baseline model is a machine learning model which uses simpler conventional methods for NLP which do not consider or include possible contextual relations inside or between cells. This model serves as a reference point to show how inclusion of context improves classification results. Initially, the method creates word embeddings using GloVe (Pennington, Socher, & Manning, 2014) for each word, which are then averaged over the whole cell value. Aside from the word embeddings, it also creates character embeddings using a one-dimensional CNN, as recommended in Vušak et al. (2021). The character embeddings are concatenated to the averaged word embeddings. After the final embedding is created, it is passed through a fully connected neural network. The label of the cell is predicted by applying a sigmoid function to the logit outputs of the neural network. For the final result, the results of all individual cells in a column are taken into consideration and averaged.

4. Calculation

CASSED uses FLAIR (Akbik et al., 2019), a Pytorch-based framework for NLP, for training and testing on the datasets of related work and on the DeSSI dataset. A distilled and uncased version of BERT (Sanh, Debut, Chaumond, & Wolf, 2020) was used with the AdamW optimizer (Loshchilov & Hutter, 2017) and a learning rate of $5 \cdot 10^{-}5$. The model was run for 20 epochs, with a mini-batch size of 16 on an NVIDIA RTX 3090 GPU. The learning process on the DeSSI dataset took around 5 hours to complete.

4.1. Evaluation metrics

Since there are multiple classes of sensitive data, the main way to determine how well our model works is to use the F1 score calculated from the confusion matrix. When detecting sensitive data, recall is more important than precision for all labels except 'Other data'. Namely, having a false negative, where the model does not find sensitive data and thus subsequent tasks do not properly remove or de-identify that sensitive data, is more damaging than it is to have a false positive, where the model detects something that is not sensitive data as sensitive, which only leads to more data being removed or de-identified in downstream tasks. Three different types of measures are used in the related work and our study. Those measures are micro, macro, and weighted. They relate to the way in which the average result over all samples and classes is calculated. In the micro average, every example contributes equally to the final result. In the macro average approach, class results are calculated independently, and each class contributes the same amount to the final result, independent of the class size. Finally, in the weighted average approach, class results are calculated separately, and the contribution of each class relates proportionally to the number of examples of the class to the final result.

Table 3
The comparison of results of CASSED and the models of related work on their datasets on metrics they provided in their research papers.

Comparison of CASSED with related work models on their datasets			
Datasets	Results		
VizNet	Metric Weighted F1	Sherlock 0.890	CASSED 0.896
Faker	Metric Weighted F1	SIMON 0.84	CASSED 0.996
WikiTables	Metric Micro F1 Macro F1	SeLab 0.51 0.66	CASSED 0.72 0.80

In this section, the results are first shown for the comparisons of a limited version of CASSED with other methods on their datasets, which are not directly related to the task of sensitive data detection. Afterwards, the results and comparisons on the DeSSI dataset are shown.

4.2. Comparisons

In order to compare our model with related work and get a basic idea of how well the machine learning part performs, we compare the performance on datasets from similar tasks of models discussed in the related work section. We use only machine learning part of our model as our rule-based methods are specifically created with sensitive data types in mind, which mostly do not occur in these datasets. Also, we omit the rule-based methods due to the NDAs, as we are not permitted to publish the exact handcrafted rule-based methods we created. Hence, in further experiments, we evaluate only the part of the model up to the post-processing step.

While we consider that in real-world applications column headers very often carry important information which can help in the detection of column labels, datasets from related work do not take them into account. This is either because they use the available column headers to generate labels for their dataset or because they consider that headers could not be created in such a way that they adequately represent real-world data. To compare our model on these datasets, we removed the part of our model which takes column headers into account and only left the input containing field values.

In Sherlock (Hulsebos et al., 2019), the dataset was created by extracting specific columns from the VizNet (Hu et al., 2019) corpus and labeling them by their column headers, thereby creating 78 possible labels. The dataset was split into a training/validation/test set using the 60/20/20 percentages division. As seen in Table 3, their method achieved a weighted F1 score of 0.890 on the test set, while our base part of the model scored slightly better, achieving a weighted F1 score of 0.896, without using column headers or additional rule-based methods.

5. Results

5.1. Training and testing on datasets from the related work

SIMON (Azunre et al., 2019) semantically classifies columns in tabular data. The main data that they used for training, as well as their initial tests, were generated from Faker (Faker, 2021) and contained nine different label types. We followed their data generation procedure and generated 10,000 columns of data, which we split according to their proposal with a 60/30/10 training/validation/test percentage division. Taking into account that the created data was automatically generated, CASSED still achieved significantly better results than SIMON. As seen in Table 3, their model produced a weighted F1 score of 0.84, while our model classified the dataset almost perfectly, with a weighted F1 score of 0.996. The performance of our model clearly shows that only using character-level features and omitting broader context negatively impacts the model's performance.

SeLaB (Trabelsi et al., 2021) uses a curated version of the WikiTables dataset from the WikiTables corpus (Bhagavatula, Noraset, & Downey, 2015), which contains data from over 1.6 million Wikipedia pages. As seen in Table 3, their model achieved a macro F1 score of 0.51 and a micro F1 score of 0.72, CASSED managed to achieve a macro F1 score of 0.66, and a micro F1 score of 0.80, thereby showing the advantages of capturing the context of the whole column.

SATO (Zhang et al., 2019) method took a subset of the Sherlock (Hulsebos et al., 2019) dataset and split it into five parts, instead of training and testing the model on the exact dataset of Sherlock. After the cross-validation was performed on those five parts, the authors very unconventionally reported their results as the average of those five test runs of the cross-validation. They did not create a separate test set which the model has not seen, thus we did not know how to properly compare our model with theirs, since we did not want to encourage further use of methods that clearly overfit. We opted to split the dataset used in SATO with a 60/20/20 split between train, validation, and test corpora, thereby taking the datasets of their cross-validation indexed as 0, 1, and 2 as training; 3 as validation; and 4 as test sets. We achieved a weighted F1 score of 0.90, which is comparable to our result on the full Sherlock dataset and is expected since SATO worked on a segment of it.

Other methods mentioned in related work, such as TaBERT (Yin et al., 2020) and TABBIE (Iida et al., 2021), deal with the same problem of embedding a database table but are used for very different downstream tasks. They focus on the tasks of question-answering or outlier detection and are not meant to classify columns of database tables, so we consider that comparing our model with theirs would probably lead to misleading information.

Lastly, SemTab (Cutrona et al., 2022) is a yearly challenge for the ontology task of knowledge base construction, which also incorporates a task where column labels need to be predicted. However, as the task was created as an ontology challenge, the authors did not provide training and test datasets for comparison, but rather tested the submitted models which were trained on arbitrary datasets, or used approaches that are not based on machine learning and did not require training data. As the experiments were conducted without a predefined training and test set, we did not see a feasible way to properly compare our model to its results.

5.2. Training and testing on the DeSSI dataset

The results on the DeSSI dataset (test part) for the CASSED model that uses only the machine learning part are presented in Table 4. They are presented for each label, giving its precision (P), recall (R), and F1 measures. The results show a very high and sometimes perfect F1 score for some classes. The test dataset results of the baseline model, related work, and both CASSED with and without the rule-based methods are reported in Table 5. The results show that CASSED significantly outperforms the baseline model, as well as Sherlock and SeLaB, in all aspects, while the difference between the CASSED models with and without rule-based methods can mostly be seen in the macro-averaged metric types. While the machine learning part on its own has very good results in general, the rule-based part of the model provides classification when a label has been found under strict rules, and in doing so only bolsters the recall of labels which can be detected in such a way. Rule-based methods are not used on the most represented labels, such as Other data, Phone number, Address, and Person, and thus the improvement will mostly be visible in the macro-averaged recall metric, and less so in other metrics.

Table 4
The results showing precision (P), recall (R), F1-score (F1), and support (S) of only the machine learning part of CASSED on the synthetically generated DeSSI dataset, test part.

Results of CASSED on DeSSI by label				
Class	P	R	F1	S
Other data	0.9955	0.9940	0.9947	1334
Phone number	0.9966	0.9954	0.9960	876
Address	0.9973	0.9987	0.9980	742
Person	1.0000	0.9852	0.9925	741
Email	1.0000	0.9956	0.9978	678
NIN	0.9793	0.9822	0.9807	673
Date	0.9982	0.9982	0.9982	570
Organization	1.0000	1.0000	1.0000	434
GPE	1.0000	1.0000	1.0000	424
Geolocation	1.0000	1.0000	1.0000	400
SWIFT/BIC	0.8667	0.9811	0.9204	53
IBAN	1.0000	1.0000	1.0000	28
Passport	0.9130	0.8400	0.8750	25
Religion	1.0000	1.0000	1.0000	22
CCN	1.0000	1.0000	1.0000	18
ID Card	0.8000	1.0000	0.8889	16
Sexuality	1.0000	1.0000	1.0000	14
Gender	0.9286	1.0000	0.9630	13
Nationality	0.9231	1.0000	0.9600	12
Race	1.0000	1.0000	1.0000	9

Table 5
The results of the baseline model, Sherlock, SeLab, CASSED with only the machine-learning part (CASSED*), and the whole CASSED model on macro-averaged and weighted metrics of precision (P), recall (R), and F1-score (F1) on the synthetically created DeSSI dataset, test part.

Comparisons of models on DeSSI						
Type	Metric	Baseline	Sherlock	SeLaB	CASSED*	CASSED
Macro	P	0.9350	0.928	0.7531	0.9699	0.9707
	R	0.8193	0.888	0.8586	0.9885	0.9978
	F1	0.8672	0.895	0.7790	0.9783	0.9832
Weighted	P	0.9205	0.933	0.7897	0.9943	0.9944
	R	0.9181	0.932	0.7360	0.9936	0.9951
	F1	0.9170	0.931	0.7429	0.9939	0.9946

5.3. Training and testing on real-world data

The principal goal of any sensitive data detection approach is to work efficiently with real-world data. Therefore, for internal testing and analyzing how well DeSSI represents real-world data, we also created another dataset from real-world data which we cannot publish due to the strict NDA. That dataset is around half the size of our dataset and the results on it are slightly worse than the results on DeSSI, by around 0.02 for weighted F1, achieving a weighted F1 score of 0.976 (compare to Table 5). The slight drop in performance is expected and can be attributed to the variations and noisy data that are found in real-world datasets. In creating DeSSI, we aimed to introduce as much nuance and noise as possible, however, due to the diverse nature of these variations in real-world datasets, it is impossible to represent all of them without making the dataset lose its generalizability, thus it is expected that results will always be slightly better on DeSSI than on real-world datasets.

6. Discussion

In this paper, an approach to the problem of classifying sensitive data types in database table columns is presented. As mentioned, the definition of sensitive data is not set in stone and varies depending on the topic. We tested our CASSED approach on a handful of other related datasets on similar tasks and outperformed the methods that use those datasets (Table 3) without relying on column headers, and by only using the machine learning part of our model. However, since these datasets were not really in the domain of sensitive data, we opted to create a dataset from synthetic and publicly available data called DeSSI which includes the most important sensitive data types. Although the dataset could certainly benefit from expansion in both its scope and the set of sensitive data types, it should provide a good starting point for comparing other models to ours for this task. Our method was also applied to a dataset created using real-world test and production databases, for which we do not have publication rights. The results on real-world datasets give insight into the possible differences and shortcomings of our synthetic dataset to be addressed in the future.

In addition to testing CASSED on datasets from related work, we also tested it on DeSSI and compared it to the performance of models described in the related work section and our baseline model, as seen in Table 5. These experiments served as a reference point to how much of an improvement active inter-cellular context can achieve.

Unlike the baseline method and the related work models which use no context, or only use passive inter-cellular context, CASSED makes the assumption that the data within the cells may be in some form of natural language and that the task of classifying a column benefits from the model being able to direct its attention to one of the cells at any given moment. To make use of this assumption, CASSED takes advantage of the extensive capabilities of BERT to account for context by converting the database tables into an input that encapsulates whole columns along with metadata and passing it to BERT for classification.

As seen in Table 5, the results show that CASSED outperforms the baseline model and the related work models in all aspects. It also achieves very high F1 scores in all sensitive data categories (Table 4), and in some which are easier to detect, even has perfect F1 scores, which indicates that the incorporation of intra-cellular context through BERT and active inter-cellular context is a viable approach that should be considered for the problem of structured sensitive data detection and, more generally, semantic column labeling.

Rule-based methods enhance the performance of CASSED to a degree (Table 5), by detecting certain labels which occur in very strict formats which are not always easily detectable for machine learning models. An argument can even be made to not classify some of the labels in the machine learning part of the model, but instead rely only on rule-based methods for these labels if they have such a strict format that the rule-based methods should almost never misclassify them.

We mentioned that CASSED has a small drop in performance on real-world data, which implies that the DeSSI dataset could use some careful expansions and adjustments, both in scope and in the number of columns, as well as discovering where the variations and noisiness in real-world data exactly come from. These expansions and adjustments should be carefully conducted in order to retain generalization and not adjust too much to one specific real-world dataset. Nevertheless, we consider that the small drop in performance only attests to the soundness and robustness of the proposed approach.

7. Conclusion

We propose a novel method, called CASSED, for solving the problem of structured sensitive data detection as well as for solving the more general semantic column labeling problem. The novelty of the method comes from the use of NLP on whole columns of structured data. Specifically, the method thereby allows the NLP model to actively consider multiple cells from the same column while creating embeddings and labeling the column. The method automates efficiently on large sets of database columns. In addition to the proposed method, a new dataset, called DeSSI, was created, to aid in the task of structured sensitive data detection and is, to our knowledge, the first widely available dataset of this type. CASSED outperforms the related work on the broader problem of semantic column labeling on their datasets, and it also outperforms the related work and a baseline model on DeSSI's structured sensitive data detection problem, which brings us to the conclusion that NLP together with the whole column context can be used to gather more information about structured data.

The potential next steps would include the incorporation of even more context in the NLP models without increasing the computational times significantly, as well as further expansion and improvement of the DeSSI dataset.

Funding

This work has been carried out within the project “Digital platform for ensuring data privacy and prevention of malicious manipulation of the personal data - AIPD2” (grant no. KK.01.2.1.02.0038), funded by the European Regional Development Fund in the Republic of Croatia under the “Operational Programme Competitiveness and Cohesion 2014–2020”.

Ethical statement

We hereby certify that all personal data used during training and testing has been obtained in strict compliance with laws and regulations protecting the rights of individuals, and that none of the personal data collected has been shared or disclosed to third parties. We further confirm that all of the data accompanying this article is either open source data or synthetic (generated) data and as such poses no risk for violation of the rights of individuals under data protection laws and regulations.

CRediT authorship contribution statement

Vjeko Kužina: Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Visualization, Writing –original draft. **Ana-Marija Petric:** Data curation, Writing – original draft, Writing – review & editing. **Marko Barišić:** Conceptualization, Software, Data curation. **Alan Jović:** Supervision, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The link to the openly available dataset is shared in the references under DeSSI 2022.

References

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., & Vollgraf, R. (2019). FLAIR: An easy-to-use framework for state-of-the-art NLP. In *2019 annual conference of the North American chapter of the association for computational linguistics (demonstrations)* (pp. 54–59).

Ateniese, G., Felici, G., Mancini, L. V., Spognardi, A., Villani, A., & Vitali, D. (2013). Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. CoRR abs/1306.4447. http://arxiv.org/abs/1306.4447.

Azunre, P., Corcoran, C., Dhamani, N., Gleason, J., Honke, G., Sullivan, D., ... Morgan, J. (2019). Semantic classification of tabular datasets via character-level convolutional neural networks. arXiv preprint arXiv:1901.08456.

Bhagavatula, C. S., Noraset, T., & Downey, D. (2015). TabEL: Entity linking in web tables. In *International semantic web conference* (pp. 425–441). Springer.

CCPA (2018). California consumer privacy act of 2018. https://cdp.cooley.com/ccpa-2018/. (Accessed 23 August 2021).

Chen, R. J., Lu, M. Y., Chen, T. Y., Williamson, D. F. K., & Mahmood, F. (2021). Synthetic data in machine learning for medicine and healthcare. *Nature Biomedical Engineering*, *5*, 493–497. http://dx.doi.org/10.1038/s41551-021-00751-8, (Accessed 3 January 2022).

Cutrona, V., Chen, J., Efthymiou, V., Hassanzadeh, O., PeJiménez-Ruiz, E., Sequeda, J., ... Oliveira, D. (2022). Results of SemTab 2021. In *Proceedings of the semantic web challenge on tabular data to knowledge graph matching*: *Vol. 3103*, (pp. 1–12). CEUR Workshop Proceedings.

DeSSI (2022). DeSSI - dataset for structured sensitive information. https://www.kaggle.com/sensitivedetection/dessi-dataset-for-structured-sensitive-information. (Accessed 14 January 2022).

Devlin, J., Chang, M. -W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Dwork, C., Smith, A., Steinke, T., & Ullman, J. (2017). Exposed! A survey of attacks on private data. *Annual Review of Statistics and Its Application*, (Accessed 8 April 2021).

Faker (2021). Faker. https://faker.readthedocs.io/en/master/. (Accessed 23 February 2021).

GDPR (2016). Regulation (EU) 2016/679 of the European parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/EC (General Data Protection Regulation). https://eur-lex.europa.eu/eli/reg/2016/679/oj. (Accessed 23 August 2021).

Google (2018). Cloud data loss prevention. https://cloud.google.com/dlp. (Accessed 10 January 2022).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780.

Hu, K., Gaikwad, S., Hulsebos, M., Bakker, M. A., Zgraggen, E., Hidalgo, C., ... Demiralp, C. (2019). VizNet: Towards a large-scale visualization learning and benchmarking repository. In *Proceedings of the 2019 CHI conference on human factors in computing systems* (pp. 1–12).

Hulsebos, M., Hu, K., Bakker, M., Zgraggen, E., Satyanarayan, A., Kraska, T., ... Hidalgo, C. (2019). Sherlock: A deep learning approach to semantic data type detection. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1500–1508).

Iida, H., Thai, D., Manjunatha, V., & Iyyer, M. (2021). TABBIE: Pretrained representations of tabular data. arXiv preprint arXiv:2105.02584.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning* (pp. 1188–1196). PMLR.

LGPD (2018). Lei geral de protecao de dados pessoais (general personal data protection law). (Accessed 4 January 2022).

Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101.

Marrero, M., Urbano, J., Sánchez-Cuadrado, S., Morato, J., & Gómez-Berbís, J. M. (2013). Named entity recognition: Fallacies, challenges and opportunities. *Computer Standards & Interfaces*, *35*, 482–489. http://dx.doi.org/10.1016/j.csi.2012.09.004, https://www.sciencedirect.com/science/article/pii/S0920548912001080.

Microsoft (2016). Microsoft power BI. https://powerbi.microsoft.com/en-us/. (Accessed 10 January 2022).

Nikolenko, S. I. (2019). Synthetic data for deep learning. CoRR abs/1909.11512. http://arxiv.org/abs/1909.11512.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing* (pp. 1532–1543).

PII (2018). PII catcher for files and databases. https://pypi.org/project/piicatcher/. (Accessed 10 January 2022).

Quinn, P., & Malgieri, G. (2021). The difficulty of defining sensitive data—The concept of sensitive data in the EU data protection framework. *German Law Journal*, *22*, 1583–1612. http://dx.doi.org/10.1017/glj.2021.79.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2020). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv:1910.01108.

Spector, A. Z., Norvig, P., Wiggins, C., & Wing, J. M. (2022). *Data science in context: Foundations, challenges, opportunities*. Cambridge University Press, http://dx.doi.org/10.1017/9781009272230.

Trabelsi, M., Cao, J., & Heflin, J. (2021). SeLaB: Semantic labeling with BERT. In *2021 international joint conference on neural networks* (pp. 1–8). http://dx.doi.org/10.1109/IJCNN52387.2021.9534408.

Trifacta (2014). Data wrangling tools & software. https://www.trifacta.com. (Accessed 10 January 2022).

Vušak, E., Kužina, V., & Jović, A. (2021). A survey of word embedding algorithms for textual data information extraction. In *2021 44th International convention on information, communication and electronic technology* (pp. 181–186). IEEE.

Wu, C., Wu, F., Qi, T., & Huang, Y. (2020). Named entity recognition with context-aware dictionary knowledge. In *China national conference on Chinese computational linguistics* (pp. 129–143). Springer.

Ye, X., Chen, Q., Wang, X., Dillig, I., & Durrett, G. (2020). Sketch-driven regular expression generation from natural language and examples. *Transactions of the Association for Computational Linguistics*, *8*, 679–694.

Yin, P., Neubig, G., Yih, W. -t., & Riedel, S. (2020). TaBERT: Pretraining for joint understanding of textual and tabular data. arXiv preprint arXiv:2005.08314.

Zhang, D., Suhara, Y., Li, J., Hulsebos, M., Demiralp, C., & Tan, W. -C. (2019). SATO: Contextual semantic type detection in tables. arXiv preprint arXiv:1911.06311.