Methods for Automatic Sensitive Data Detection in Large Datasets: a Review

Vjeko Kužina, Eugen Vušak, Alan Jović
University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
{vjeko.kuzina, eugen.vusak, alan.jovic}@fer.hr

***Abstract*—In recent years, the need for detection and de-identification of sensitive data in both structured and unstructured forms has increased. The methods used for these tasks have evolved accordingly and currently there are many solutions in different areas of interest. This paper describes the need for the detection of sensitive data in large datasets and describes the challenges associated with automating the detection process. It gives a brief overview of the rule-based and machine learning methods used in this area and examples of their application. The advantages and disadvantages of the described methods are also discussed. We show that the most recent detection solutions are based on the latest and most advanced models proposed in the field of natural language processing, but that there are still some rule-based methods used for certain types of sensitive data. In recent years, the need for detection and de-identification of sensitive data in both structured and unstructured forms has increased. The methods used for these tasks have evolved accordingly and currently there are many solutions in different areas of interest. This paper describes the need for the detection of sensitive data in large datasets and describes the challenges associated with automating the detection process. It gives a brief overview of the rule-based and machine learning methods used in this area and examples of their application. The advantages and disadvantages of the described methods are also discussed. We show that the most recent detection solutions are based on the latest and most advanced models proposed in the field of natural language processing, but that there are still some rule-based methods used for certain types of sensitive data.**

***Keywords*—sensitive data, detection, de-identification, unstructured data, machine learning, named entity recognition.**

I. Introduction

Since the emergence of the Internet, and even more so in the last decade, the amount of data available to everyone has increased dramatically. The ability to share and process vast amounts of data is a great catalyst for the world's research and development, but within vast amounts of data lies a significant amount of sensitive information that can be misused if not handled properly. The misuse of such data is prohibited by law. As there has been more and more processing of big data, and more and more leaks of databases containing sensitive information over time, the need for data loss prevention has become even greater. Sensitive personal data, or sensitive data for short, represents personal information that by its nature is particularly sensitive in terms of fundamental rights and freedoms [1]. As such, the processing of sensitive data is prohibited unless certain conditions are met or permissions are granted. In the medical domain, medical records can be very useful to gain knowledge or find yet unknown relations through machine learning, but these records cannot be published as they contain sensitive data, which should be found and removed, anonymized or pseudo-anonymized beforehand. In the financial domain, companies need to be able to find sensitive data of their clients and remove the data at the request of a client or when an event occurs that requires such actions. These two domains come to the same problem of detecting sensitive data from two different perspectives. The problem can be solved manually by having people annotate each occurrence of sensitive information. Although this is a relatively secure solution, it is very expensive, slow and time-consuming. A much better solution would be to automate the process of detecting sensitive data. Sensitive data can come in both structured and unstructured forms, making it even more difficult to detect automatically. The described problems are collectively widely known as the sensitive data de-identification problem, which consists of two parts. In the first part, the sensitive data must be detected, and in the second part, it must be removed, anonymized, or pseudo-anonymized. This paper addresses the first problem by presenting and reviewing methods to detect sensitive data.

The task of sensitive data detection in unstructured text is essentially a special case of the Named Entity Recognition (NER) task, which would constitute the detection of certain entities that represent types of sensitive data. While sensitive data detection in structured data depends entirely on the domain and its associated metadata, the recent implementations of such tasks use machine learning on pre-trained word embeddings ([2], [3]).

In the following sections, we first review the related works that have compared or analyzed different methods for sensitive data detection. Then, we list the most commonly used methods, why these methods are or have been used for sensitive data detection, and how these methods have evolved. Finally, the paper will talk about the good and bad sides of these methods, compare them and give a conclusion on what is the current trend of development for the problem and why that is the case.

II. Related Work

Several studies have been done describing the problem of de-identification in specific domains and comparing the methods used to solve the problem.

Stubbs et al. [4] created a dataset for de-identifying medical records and held a competition to detect sensitive data in the created dataset. They compared and presented the different approaches of the submissions to the competition and concluded that the best solutions at the time used combinations of machine learning and hand-crafted methods for detecting sensitive data. Leevy et al. [5] compared different long short-term memory network (LSTM) and conditional random field (CRF) approaches to the problem of de-identifying sensitive data in medical text. Most of the studies in the survey were conducted on different tasks as they either used different datasets or predicted different subsets of entities of sensitive data. LSTM and CRF approaches outperformed each other on approximately the same number of tasks, resulting in no clear winner between the two approaches. Garfinkel [6] compared different rule-based and machine learning approaches for the de-identification problem in medical documents. They found that rule-based systems are better for entities that have a strict form, such as zip codes or social security numbers, but are generally much worse than methods based on machine learning. Meystre [7] examined different kinds of de-identification methods and discussed the strengths and weaknesses of rule-based and machine learning methods in general for de-identifying clinical records and provided examples of systems that use these methods. Trienes et al. [8] created a dataset for de-identification of medical records in which they compared three methods. The rule-based method had the worst performance and cannot be generalized to different domains. The feature-based model which uses CRF performed in the middle for the main problem as well as in different domains, while the neural network approach using a bidirectional LSTM (BI-LSTM) along with a CRF performed best overall. Truong et al. [9] address the problem of sensitive data detection in the financial domain, which has not been thoroughly researched because of the lack of publicly available datasets. They created their own datasets, and evaluated several approaches on that dataset, with Convolutional Neural Networks performing the best amongst all the methods.

There also exists research for the detection of sensitive data in formats which do not directly involve text. Some of them use some kind of pre-processing to transform the original data into text, after which the same methods are used for the detection of sensitive data in structured or unstructured text, others employ entirely different techniques. For example, Google Cloud DLP [10] uses Optical Character Recognition (OCR) to transform images of textual documents into text, as well as their speech-to-text API for transforming audio files to text. Other possible approaches can detect biometric data in images and videos with face-recognition techniques and as such be useful in sensitive data detection.

There are software products, as shown in Table I, which tackle the problem of text de-identification within the problem of Data Loss Prevention (DLP), and they approach the problem of detecting sensitive data on a part of, or on all of unstructured and structured text, as well as on images pre-processed with OCR, using methods based on rules or machine learning.

The field of sensitive data detection in structured data has not been thoroughly researched. Some products mentioned in Table I offer tools for this task, but they do not go into more detail than using rules and machine learning. We could not find any scientific work focusing on automatic detection of sensitive data in structured data or similar topics, which suggests that the topic is currently largely unexplored. Therefore, most of the paper will be related to the detection of sensitive data in unstructured data.

III. Overview of Approaches

The problem of sensitive data detection can be approached in a number of ways, but most of these methods fall into one of two categories. The first category is rule-based methods, and the second is machine learning methods. The final solution to the problem can also be a combination of several approaches used together for different parts of the task, or one approach can be the input to another approach.

A. Rule-based approaches

Rule-based approaches describe rules that decide what the model recognizes as sensitive, and what it recognizes as not sensitive. These rules are created by people who have a deep understanding of the domain and the rules take lot of time and resources to create.

1) Lookup table: Some of the first approaches used for sensitive data detection, but mostly not used on their own, were lookup tables. The idea behind this is to create a hash table of frequently used terms that potentially tells us whether a word is sensitive or not. Lookup tables can be used to detect both sensitive and non-sensitive data. In the first approach, lookup tables contain words that are often or always entities. For example, there might be a lookup table that contains the most common first names, and another lookup table that contains the most common last names. In the second approach, which could be used when lookup tables are used in combination with another method, the lookup table could consist of words that are not sensitive, but are often recognized as sensitive by another method, which would remove the misclassification. Furthermore, if the given data is unstructured, the surrounding words such as “Mr.” or “Dr.” may be used to detect potentially sensitive words near them. Therefore, creating a lookup table with such indicators would allow detection of sensitive words that are not in lookup tables but are near the indicators.

Depending on the task at hand, specific lookup tables can be created to include the domain of the task. Some medical records’ implementations used a list of names of staff members, patients, or recently deceased individuals ([16] [17] [18]), and some used names of institutions such as hospitals or clinics ([19]), along with the lists of the most common names in each country or similar common knowledge.

2) Regular expressions: Regular expressions represent a search pattern defined by a sequence of characters. If a pattern is found in sensitive data that does not occur in non-sensitive data, then regular expressions are the way to go as they find

TABLE I
DE-IDENTIFICATION SOFTWARE PRODUCTS

Product	Data use cases	Approach used
Google Cloud DLP [10]	Audio, Images, Unstructured and Structured text	Machine learning, Rule-based, and OCR
IBM Security Guardium [11]	Structured text	Rule-based
Nightfall AI [12]	Images, Unstructured and Structured text	Machine learning and OCR
Gretel AI [13]	Unstructured and Structured text	Machine learning and Rule-based
Presidio [14]	Images and Unstructured text	Machine learning, Rule-based, and OCR
PII Catcher [15]	Structured text	Machine learning and Rule-based

all occurrences of that pattern and thus detect only sensitive words.

Much of the sensitive data such as dates, identification numbers, email addresses, etc. follow a pattern or must be in one of several possible formats in which that type of data occurs. For example, while identification numbers always have a strict format, dates occur in multiple formats, but a regular expression that takes all of these formats into account at the same time can easily be created, thus identifying all occurrences of these patterns as sensitive data. Beckwith et al. [16], Friedlin and McDonald [17], and Neamatullah et al. [19] used regular expression for, among other things, address, location, or email patterns, as well as for detecting words consisting mainly of digits.

3) Identifying metadata: Metadata is data that gives information about other data that is being used. In structured data, metadata is always present because the structure itself conveys some information. If the metadata can be helpful in recognition, then it is also identifying. For example, metadata would be the name of a column in a database table or the name of a section in a form. If the metadata conveys useful information for our task such as the name of the "First name" column, then that would mean that the data present in that column is sensitive. Beckwith et al. [16] used identifying information from XML document headers that contained names and dates, along with regular expressions, to find sensitive data and remove it from all existing locations in the document.

B. Machine learning approach

Machine learning methods use various algorithms to train themselves to recognize patterns without having to explicitly communicate these patterns to the algorithm. Since some of the most basic machine learning algorithms only consider the currently observed input, or word in our case, many works have not used them, but rather algorithms that in some way capture the context surrounding the currently observed word, since the problem of detecting sensitive data in unstructured data depends on context.

1) Hidden Markov model: Hidden Markov models (HMM) [20] are categorized as generative models that use latent variables (hidden states) representing entities (outputs) to predict observable variables (inputs). The hidden states are interconnected and have probabilities of transitions from one to another as well as probabilities of producing a particular input. The model maximizes the joint probability for the entire sequence of tags along with the entire input sequence, rather than for a single tag, because in this way previous words and tags change the classification of subsequent tags. Chen et al. [21] used HMMs on data introduced by Stubbs et al. [4]. In the preprocessing part, each word was embedded into a vector and this embedding was further given as input to the model. The model was allowed to use as many hidden states as the data itself dictates by using the latent Dirichlet process [22]. This allowed the model to capture variations in the data and thus create more distinct categories. For example, the word "a" by itself is not a sensitive word and usually suggests that the word to come is not sensitive either, but if the words "works" and "as" are present before the word "a" then it suggests that the next word will be a sensitive word that would represent an occupation.

2) Conditional random fields: CRFs [23] are a generalization of HMMs, they follow the same idea of hidden states except that the states are undirected, which allows the model to use information from both previous and subsequent inputs as well as possibly other features represented as hidden states. The most important difference between CRFs and HMMs is that CRFs are discriminative rather than generative models, because they maximize the conditional probability of outputs given inputs, whereas HMMs maximize the joint probability of inputs and outputs co-occurring. These differences allow the CRF model to create arbitrary features that need not be statistically independent and are not restricted to modeling dependencies of hidden states and their associated observations. These arbitrary features are often handcrafted and specific to the domain. In the field of sensitive data detection, they are often created by rule-based methods since the rule-based methods are better at detecting certain kinds of sensitive data.

Implementations of CRFs for the task of de-identification use different features to try to predict the most likely labels. For example, Berg and Dalianis [24] used lemmas, first few and last few letters of words, and binary and integer indicators, among others. If the word consists only of numbers, the binary indicator would be a "1", and if not, then a "0". Similarly, the integer indicator could indicate how many letters are in the word. Liu et al. [25] used one CRF for various token-level features such as Bag of Words, part-of-speech (POS) tags, and orthographic features, and another CRF for character-level features such as Bag of Characters, which used unigrams, bigrams, and trigrams, as well as sentence information.

3) Recurrent neural networks: Recurrent neural networks (RNNs) are types of neural networks that contain an internal state (latent variable) that is modified by inputs and produces outputs. The state thus acts as a kind of memory that allows past words to influence future output decisions. Like the HMM, RNNs also model the distribution of a sequence of observations from latent variables, but RNNs have one latent variable that is changed by each input that comes to it, while HMMs have multiple latent variables that are not changed by the inputs, but only transitions between each other using previous states and the current input. Srivastava et al. [26] used two types of RNNs, the first of which generated the new internal state from the previous internal state and the current input, while the second RNN used the output of the previous internal state along with the input of the current state to generate the new internal state. The RNN's input was an embedding of the target word and its surrounding words to better capture short-term temporal dependencies.

4) Long short-term memory: LSTM is a modification of an RNN that facilitates recall of previous input, and solves some of the problems RNNs have faced when processing long sequences, such as vanishing or exploding gradients. LSTM has the same general architecture as an RNN, the only difference is that the internal state (memory) is more complex. It uses several matrices represented in the form of gates. The first gate decides which part of the input modifies the memory, the second decides which parts of the memory are forgotten, and the third gate decides which parts of the memory are used to generate the output. Implementations of LSTMs for the de-identification task mostly use a BI-LSTM consisting of two LSTMs, the first of which trains on the sequence as it normally is, and the second on a sequence with a reversed order of words.

Richter-Pechanski et al. [27] used a BI-LSTM with a concatenation of character-level word embeddings and embeddings obtained from ELMO (Embeddings from Language Models [28]), a word representation model trained on large amounts of unlabeled data. Madan et al. [18] also used a BI-LSTM, but with character-level embeddings concatenated with POS tag embeddings.

5) BERT: BERT [29] is a recent deep learning model that uses attention through bidirectional transformers [30] to capture important features in natural language. It allows the model to consider the entire input while predicting each output, and the model trains itself on which part of the input to pay the most attention to. The model is pre-trained on huge amounts of unlabeled data using a masked language model, laying a good foundation for transfer learning to a variety of different domains.

Garcia-Pablos et al. [2] and Johnson et al. [3] used BERT for the task of sensitive data de-identification. They tokenized their sentences as inputs to a pre-trained BERT and refined it with a fully connected linear layer that has the outputs of BERT as inputs, and the log-likelihood of the classes as outputs.

IV. Discussion and Application Examples

Over the years, as seen in the previous section, there have been many approaches to the problem of sensitive data detection in unstructured text, most of which fall into two categories: rule-based methods and machine learning methods. Both categories have different advantages and disadvantages.

Rule-based methods use domain knowledge to create patterns that are recognized by the system. They require very few or no training examples because they represent patterns or rules and the model itself does not need to learn from examples. It is also easy to add new rules or implement special cases as the need arises. However, since rule-based methods are based on hand-crafted rules, it also means that all specific and rare cases must be considered, and complex solutions must be created for some of these cases. Moreover, the engineers behind the methods need to know all the edge cases and possible scenarios that can occur in order to build a good model. Another disadvantage of rule-based methods is that once a system is created, it has very low generalizability because it was built specifically for the problem at hand, and if the domain of the entities being searched for changes even slightly, it cannot adapt to them without a lot of work.

Machine learning methods themselves do not take domain knowledge into account, as they require large amounts of annotated data to learn to work properly, which usually requires a lot of work from experts in the domain. If the model does not detect a particular case where sensitive data occurs, it is very difficult to make a small change and thereby include that edge case. Considering all these drawbacks, machine learning models are still generally better than rule-based methods because they do not require expert knowledge and manual work in creating the rules. This means that they do not need to know all the edge cases, because the model learns them by itself if it is good enough and if there is enough annotated data. Machine learning methods also have very high generalizability because they do not need completely new rules for a slightly different domain, but can be trained on the data of the new domain and learn to adapt to it.

Although these two approaches seem very different and have different strengths, the best models usually incorporate both approaches. There are two ways to combine these approaches. The first is to leave the detection of some entities or certain instances of entities to rule-based methods, e.g., social security or phone numbers, while the machine learning algorithm detects the other entities. The second approach is to use rule-based methods to generate features that are used by machine learning algorithms along with the words from the text, e.g., a feature could be 1 if the current word consists only of numbers and 0 if not, helping the machine learning model recognize that a word could represent a phone or social security number. As shown in [4] and [8], the best performers in sensitive data detection tasks were machine learning algorithms using rule-based features in machine learning algorithms, followed by the methods that used machine learning algorithms for some entities and rule-based methods for others, followed by solely

machine learning algorithms, and finally, purely rule-based methods had the worst performance.

LSTMs and CRFs have shown the best results in several studies, as shown in [5]. LSTMs tend to perform better than CRFs, but when CRFs are combined with rule-based feature extraction systems, there does not seem to be a clear winner currently, as both perform slightly better than their counterpart on some tasks, but not on others. The most recent studies [2], [3] have exploited the novelty of BERT in sensitive data detection after having produced several state-of-the-art results in a variety of NLP tasks. Although there are not many such models yet, the studies conducted have yielded promising results hinting at the direction in which automated sensitive data detection is moving and where future work is needed.

In Table II, we provide results of different methods on several de-identification datasets. On the i2b2 clinical naratives de-identification challenge [4] dataset, BERT achieved the best result followed by a combination of CRF and LSTM. Other machine learning approaches have a somewhat similar F1 score, while only rule-based approaches performed significantly worse. However, the approaches that use fewer entity types may not be directly comparable to the approaches that use all 18. Most approaches referenced in the paper used datasets that are not widely used, or were used only in their paper, and thus cannot be directly compared to all other models, but only to the models tested on the same dataset. Garcia-Pablos et al. [2] trained and tested their BERT model on data from the Spanish text de-identification challenge MEDDOCAN [33] and ranked second, directly behind the LSTM of Lange et al. [32]. Trienes et al. [8] created a dataset of Dutch medical records called NUT for training and testing their models and found that the LSTM performed the best with an F1 measure of 91.6%, with the CRF model close behind and the rule-based approach coming in last. They also acheieved an F1 score of 91.2% on the i2b2 dataset.

V. Conclusion

In this paper, we have presented the problem of de-identification of sensitive data and the need for its automation in big data. We have given an overview of the used methods, divided them into rule-based and machine learning approaches, and given the reasons for their use as well as examples of where and why they might be a good choice. Furthermore, we have compared the methods and discussed their advantages and disadvantages. We have shown that the best solutions are based on the latest and most advanced methods proposed in the machine learning field, but that many approaches still use rule-based methods in some aspects. Finally, there is still a need for further research on the latest methods and algorithms for automatic detection of sensitive data.

Acknowledgment

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", funded by the European Regional Development Fund in the Republic of Croatia under the "Operational Programme Competitiveness and Cohesion 2014 – 2020."

References

[1] European Parliament and Council of European Union, *Regulation (eu) 2016/679*, https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679, 2016.

[2] A. Garcıa-Pablos, N. Perez, and M. Cuadros, "Sensitive data detection and classification in spanish clinical text: Experiments with BERT," *arXiv preprint arXiv:2003.03106*, 2020.

[3] A. E. Johnson, L. Bulgarelli, and T. J. Pollard, "Deidentification of free-text medical records using pre-trained bidirectional transformers," in *Proceedings of the ACM Conference on Health, Inference, and Learning*, 2020, pp. 214–221.

[4] A. Stubbs, C. Kotfila, and Ö. Uzuner, "Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/uthealth shared task track 1," *Journal of Biomedical Informatics*, vol. 58, S11–S19, 2015.

[5] J. L. Leevy, T. M. Khoshgoftaar, and F. Villanustre, "Survey on rnn and crf models for de-identification of medical free text," *Journal of Big Data*, vol. 7, no. 1, pp. 1–22, 2020.

[6] S. L. Garfinkel, "De-identification of personal information," *NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD, USA*, no. 8053, 2015. DOI: 10.6028/NIST.IR.8053.

[7] S. M. Meystre, "De-identification of unstructured clinical data for patient privacy protection," in *Medical Data Privacy Handbook*, Springer, 2015, pp. 697–716.

[8] J. Trienes, D. Trieschnigg, C. Seifert, and D. Hiemstra, "Comparing rule-based, feature-based and deep neural methods for de-identification of dutch medical records," *Proc. 1st ACM WSDM Health Search and Data Mining Workshop (HSDM2020)*, 2020.

[9] A. Truong, A. Walters, and J. Goodsitt, "Sensitive data detection with high-throughput neural network models for financial institutions," *arXiv preprint arXiv:2012.09597*, 2020.

[10] *Cloud Data Loss Prevention*, https://cloud.google.com/dlp, Accessed: 2021-01-18.

[11] *IBM Security Guardium Data Protection*, https://www.ibm.com/products/ibm-guardium-data-protection, Accessed: 2021-05-05.

[12] *Nightfall AI*, https://nightfall.ai/, Accessed: 2021-05-05.

[13] *Gretel AI*, https://gretel.ai/, Accessed: 2021-05-05.

[14] *Presidio - Data Protection and Anonymization API*, https://github.com/microsoft/presidio, Accessed: 2021-05-05.

[15] *PII Catcher for files and Databases*, https://pypi.org/project/piicatcher/, Accessed: 2021-01-18.

TABLE II
PERFORMANCE OF DIFFERENT DE-IDENTIFICATION METHODS ON THE I2B2, MEDDOCAN, AND NUT DATASETS

Dataset	Approach used	Methods used	No. of entity types	F1,%	Reference
i2b2	Machine learning	BERT (pre-trained embeddings)	18	98.62	Johnson et al. [3]
i2b2	Machine learning and rule-based	CRF + LSTM + hand-crafted features	18	96.98	Liu et al. [25]
i2b2	Machine learning	LSTM (character level and POS tag embeddings)	4	95.92	Madan et al. [18]
i2b2	Machine learning and rule-based	CRF + hand-crafted features	18	95.10	Bui et al. [31]
i2b2	Machine learning	Jordan-type RNN	7	93.84	Srivastava et al. [26]
i2b2	Machine learning	LSTM + CRF	18	91.2	Trienes et al. [8]
i2b2	Machine learning	HMM (latent Dirichlet process)	18	75.49	Chen et al. [21]
MEDDOCAN	Machine learning	LSTM	21	97.0	Lange et al. [32]
MEDDOCAN	Machine learning	BERT (pre-trained embeddings)	21	96.7	Garcia et al. [2]
NUT	Machine learning	LSTM + CRF	18	89.3	Trienes et al. [8]
NUT	Rule-based	Hand-crafter features	18	66.4	Trienes et al. [8]

[16] B. A. Beckwith, R. Mahaadevan, U. J. Balis, and F. Kuo, "Development and evaluation of an open source software tool for deidentification of pathology reports," *BMC Medical Informatics and Decision Making*,

[17] F. J. Friedlin and C. J. McDonald, "A software tool for removing patient identifying information from clinical documents," *Journal of the American Medical Informatics Association*, vol. 15, no. 5, pp. 601–610, 2008.

[18] A. Madan, A. M. George, A. Singh, and M. Bhatia, "Redaction of protected health information in ehrs using crfs and bi-directional lstms," in *7th Int. Conf. Reliability, Infocom Technologies and Optimization (ICRITO)*, IEEE, 2018, pp. 513–517.

[19] I. Neamatullah, M. M. Douglass, H. L. Li-wei, A. Reisner, M. Villarroel, W. J. Long, P. Szolovits, G. B. Moody, R. G. Mark, and G. D. Clifford, "Automated de-identification of free-text medical records," *BMC Medical Informatics and Decision Making*, vol. 8, no. 1, p. 32, 2008.

[20] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state markov chains," *The Annals of Mathematical Statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.

[21] T. Chen, R. M. Cullen, and M. Godwin, "Hidden markov model using dirichlet process for de-identification," *Journal of Biomedical Informatics*, vol. 58, S60–S66, 2015.

[22] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[23] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," *Proc. 18th Int. Conf. Machine Learning*, ICML '01, pp. 282–289, 2001.

[24] H. Berg and H. Dalianis, "Augmenting a de-identification system for swedish clinical text using open resources and deep learning," in *22nd Nordic Conference on Computational Linguistics (NoDaLiDa), Turku, Finland, September 30, 2019*, Linköping University Electronic Press, 2019, pp. 8–15.

[25] Z. Liu, Y. Chen, B. Tang, X. Wang, Q. Chen, H. Li, J. Wang, Q. Deng, and S. Zhu, "Automatic de-identification of electronic medical records using token-level and character-level conditional random fields," *Journal of Biomedical Informatics*, vol. 58, S47–S52, 2015.

[26] A. Srivastava, A. Ekbal, S. Saha, P. Bhattacharyya, *et al.*, "A recurrent neural network architecture for de-identifying clinical records," in *Proc. 13th Int. Conf. on Natural Language Processing*, 2016, pp. 188–197.

[27] P. Richter-Pechanski, A. Amr, H. A. Katus, and C. Dieterich, "Deep learning approaches outperform conventional strategies in de-identification of german medical reports.," in *GMDS*, 2019, pp. 101–109.

[28] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. 2018 Conf. NAACL*, New Orleans, Louisiana: ACL, Jun. 2018, pp. 2227–2237.

[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *Proceedings of NAACL-HLT 2019*, pp. 4171–4186,

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[31] D. D. A. Bui, D. T. Redden, and J. J. Cimino, "Is multiclass automatic text de-identification worth the effort?" *Methods of Information in Medicine*, vol. 57, no. 04, pp. 177–184, 2018.

[32] L. Lange, H. Adel, and J. Strötgen, "NLNDE: The neither-language-nor-domain-experts' way of spanish medical document de-identification," *Proc. Iberian Languages Evaluation Forum (IberLEF 2019)*, 2020.

[33] M. Marimon, A. Gonzalez-Agirre, A. Intxaurrondo, H. Rodriguez, J. L. Martin, M. Villegas, and M. Krallinger, "Automatic de-identification of medical texts in spanish: The meddocan track, corpus, guidelines, methods and evaluation of results.," in *IberLEF@ SEPLN*, 2019, pp. 618–638.