Leveraging Contextual Text Representations for Anonymizing German Financial Documents

David Biesner[*†‡], Rajkumar Ramamurthy[*†], Max Lübbering[†], Benedikt Fürst[§], Hisham Ismail[§], Lars Hillebrand[†‡], Anna Ladi[†], Maren Pielka[†], Robin Stenzel[†], Tim Khameneh[§], Vanessa Krapp[§], Ilgar Huseynov[§], Jennifer Schlums[§], Uwe Stoll[§], Ulrich Warning[§], Bernd Kliem[§], Christian Bauckhage[†], Rafet Sifa[†]

[†]Fraunhofer IAIS, Germany
[‡]University of Bonn, Germany
[§]PriceWaterhouseCoopers GmbH WPG, Germany

Abstract

Despite the high availability of financial and legal documents they are often not utilized by text processing or machine learning systems, even though the need for automated processing and extraction of useful patterns from these documents is increasing. This is partly due to the presence of sensitive entities in these documents, which restrict their usage beyond authorized parties and purposes. To overcome this limitation, we consider the task of anonymization in financial and legal documents using state-of-the-art natural language processing methods. Towards this, we present a web-based application to anonymize financial documents and also a large scale evaluation of different deep learning techniques.

Introduction

With the increasing availability of digital financial and legal documents, the demand for processing them automatically to extract patterns and to assist the users is of significant importance [Sifa et al.2019]. However, usually such financial data cannot be processed or shared beyond authorized parties due to the prevelance of sensitive information regarding certain individuals and organizations. This limits the development of machine learning tools which usually requires providing data access to researchers and developers within that organization. One possible solution is to perform either pseudo-anonymization or full anonymization of data before further processing.

In addition, with the introduction of the General Data Protection Regulation (GDPR),[1] personal data can only be further processed if they are compatible with the very strict purposes permitted by law for which this data were collected.[2] These purposes usually do not include the usage of the collected data for the training of machine learning tools. In fact, the GDPR does not even mention the processing of "Big Data" or algorithms with a single word. [Gola et al.2017] This does not change with the 2019s entry into force of a new regulation of the EU on the free flow of non-personal data. As the name already suggest, this regulation allows the storage and processing of data across the Member States without unjustified restrictions, as long as the data are not personal. However, the principle of purpose limitation is not applicable once the data are anonymized[3] and therefore this data can be used for developing digital solutions across Europe.

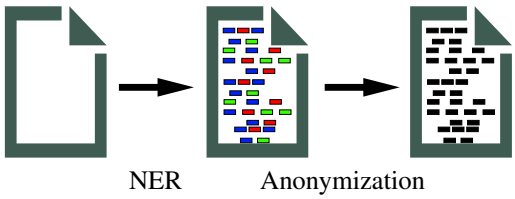

Figure 1: General workflow for anonymizing a document using named entity recognition. First, sensitive entities are identified using deep learning methods and rule-based post-processing. Then the identified entities are replaced with appropriate tags to preserve the text structure or hidden behind a general anonymized tag.

Furthermore, if the personal data are no longer necessary for the purpose for which it was collected, the GDPR grants the data subject a right to be forgotten, i.e. the right that its data are being erased[4]. In practice, a company that collects personal data, like every service provider, would need to delete their customer contracts at the time of its termination date. However, this could contradict legal retention periods, for example for tax purposes. This may be avoided, if the company anonymizes their contracts at the termination date. Considering the amount of the corresponding documents, manual anonymization is not appropriate under these circumstances.

However, the demand for anonymization of confidential data has always been present, not only since the introduction of the GDPR. For instance, publication of judgments in the public interest is, at least in Germany, a direct constitutional task for the judicial power and therefor for ev-

[*]First authors, equal contribution

[1]https://gdpr-info.eu/

[2]Art. 17 GDPR

[3]Recital 26 GDPR

[4]Art. 5 GDPR

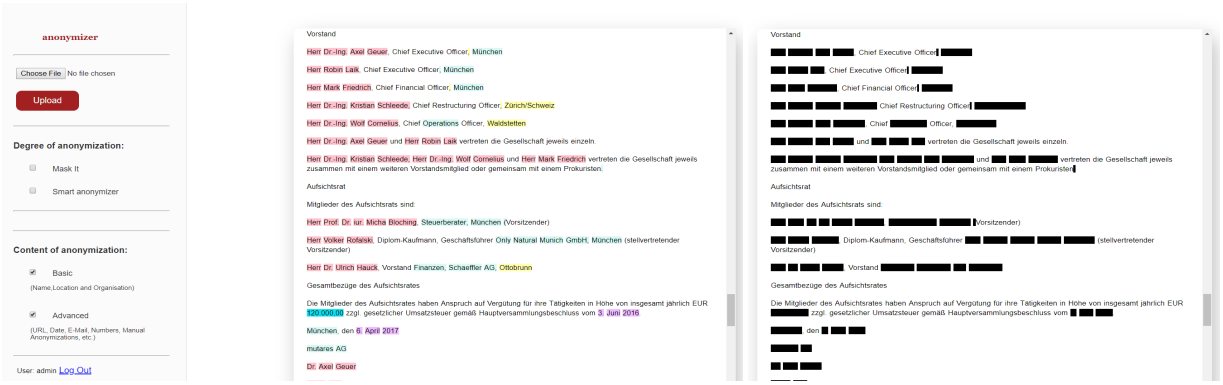

Figure 2: A screenshot of our anonymization tool; The left pane contains the UI controls for uploading the document and other settings such as to turn on the anonymization for numbers and to enable masking. To the right of it, there is the document pane and it shows the content of the document in which sensitive entities are highlighted if the mask option is not selected. If the mask option is selected, then the document pane shows the same content instead with sensitive entities masked.

ery single court[5]. However, these publications need to be anonymized, regardless of the GDPR, to protect the fundamental right to informational self-determination[6]. Until now, such anonymization is mainly done manually, resulting in a publication of only a mere fraction of the judgements that are in the public interest.

All the examples above have in common that the data with the need for anonymization is usually part of documents like contracts or other reports. Consequently, we address this concern of data privacy and protection and present a web-based anonymization application that anonymizes sensitive information such as names of persons, locations, organizations, numbers, telephone numbers, dates, and URLs in a piece of writing by the example of financial documents. We tackle this using state of the art deep learning and natural language processing techniques as well as rule based post-processing. A general outline of the workflow is shown in Figure 1.

Our main contributions in this work are:

- a method to anonymize 99% of all sensitive entities contained in German financial documents while maintaining high readability and preserving the structure of the given text
- presenting a web-based application and an API to use our method on various types of documents and
- a quantitative evaluation of multiple state-of-the-art deep learning techniques for anonymization as well as the impact of domain-specific language models for financial documents.

Related Work

Earlier systems on anonymization focused primarily on medical records. The first anonymization system was developed by [Sweeney1996] used several pattern matching algorithms which detect names, phone numbers etc. Later in 2006, a challenge was hosted to anonymize clinical data which were also made available as public dataset namely i2b2[7] for de-identification. Several systems were developed as a result of this challenge which tackled the problem using named-entity recognition [Wellner et al.2007, Gardner and Xiong2008], rule-based systems [Neamatullah et al.2008] and hybrid system [Ferrández et al.2012] which uses look-up on dictionaries, regular expressions and as well as model-based classifiers. To the best of our knowledge, we present the first large scale of evaluation of anonymization techniques with respect to financial documents.

Application

Web-based Application

A screenshot of the application is shown in Figure 2. It is a web-based application which allows the user to upload text documents (e.g. docx) and visualize the anonymized content. The interface contains two panes; a left pane with controls and a right pane where the anonymized document is rendered. There are two basic configurable settings: by default, names, locations, organizations and other entities are anonymized using our deep learning methods. Additionally one can enable anonymization of numbers, dates etc. which are detected using regular expressions. The sensitive entities are highlighted in different colors based on their types; In Figure 2, the names of person, company, location are highlighted in red, green and blue respectively. Further, the tool allows the user to enable masking such that sensitive entities are blacked out entirely as shown in the figure on the right most pane. Once the document is anonymized, the tool enables the user to download the processed document which is free from sensitive entities.

API

Since the main application of this tool is document preprocessing for further distribution or use in the training of machine learning systems, we desire the anonymization of

[5]BVerwG, 26.2.1997 6 C 3/96

[6]Art. 2 Abs. 1 GG in conjunction with Art. 1 Abs. 1 GG

[7]https://www.i2b2.org/

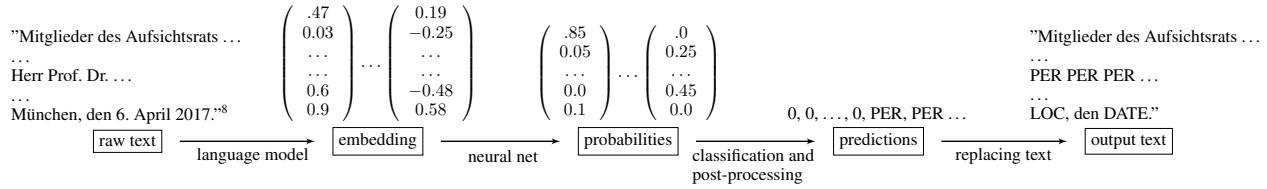

Figure 3: Workflow from raw text to final anonymized output. We convert each token into a numerical vector using a trained language model, use a neural net classifier to predict probabilities for each class for each token, choose the class with the highest probability, apply post-processing and finally replace named entities with corresponding labels in text, leaving words classified as *0* intact.

an entire document corpus. These anonymized documents can afterwards be handled by developers without clearance for the original data. For that reason, we also provide a REST API and python package for internal usage. This makes it possible for an employee with the required clearance for the original documents and no involvement in the development process to use the tool to anonymize a corpus of documents at once and return the anonymized data. This leaves a readable text without sensitive information that can be further analyzed by different machine learning approaches.

Methods

Anonymization as Sequence Tagging

We tackle the problem of anonymization as a Sequence Tagging task [Nguyen and Guo2007]. Given a document consisting of several sentences in which each sentence is a sequence of words (tokens), our goal is to assign a suitable label to each token indicating if it contains sensitive information or not. The possible labels include *0* (contains non-sensitive information), *ORG* (if it contains an organisation or part of an organisations name), *PER* (if it contains a person or part of a persons name), *LOC* (if it contains a location or part of a location name), *PROD* (if it contains a product name), *SEG* (if it contains information about the industry of the company), *URL* (if it contains an URL), *TEL* (if it contains a phone number), *DATE* (if it contains a date), *NUM* (if it contains a number), *EMAIL* (if it contains an e-mail address) and *OTH* (if it contains any other sensitive information). In particular, we refer to *ORG*, *PER*, *LOC*, *PROD*, *SEG* and *OTH* as *named entities* as it is part of the well-known problem of named entity recognition [Li et al.2018] in natural language processing.

We employ a multi step approach as depicted in Figure 3. **Step 1**: Predict the named entities in each document using language models and deep learning methods, **Step 2:** Make these predictions consistent across each document, **Step 3:** For the tokens which are not labeled yet, predict other labels using rule-based post-processing steps, **Step 4:** Replacing the text of tokens by appropriate tags for preserving the sentence structure and semantics.

[8]"Members of the board ... Prof. ... Munich, April 6th 2017."

Contextual Language Models

Unlike traditional string based methods (e.g. rule-based systems using *regex*), modern deep learning approaches for text classification require a two step approach: First, the raw text has to be converted into a numeric representation, usually a vector of fixed dimension for each word in the text. This task is done by employing a *language model*. The numeric representation of a token is then fed into a classifier that outputs probabilities for each class.

In our work, we utilize *flair* [Akbik, Blythe, and Vollgraf2018] which is a language model developed by Zalando Research. It employs a bi-directional character-based recurrent neural net that traverses each sentence in both forward and backward direction. The corresponding hidden states of the network at the beginning and end of each token together act as the numeric vector representation for that token, that contains both information on the word itself, as well as an encoding of the surrounding words, thereby capturing the context of the token. In practice, this means that the token *Vogel* in the following two sentences:

- "Herr Vogel ist Geschäftsführer der Test GmbH."[9]
- "Der frühe Vogel fängt den Wurm."[10]

will have different representation that allows the prediction layer to differentiate between the name and the animal *Vogel*. This differentiates contextual language models from word-vector models like word2vec [Mikolov et al.2013] and glove [Pennington, Socher, and Manning2014], which assign each word a global vector representation.

Apart from the training data, the major differentiating factor between the language models presented in this paper is their size, referring to the dimension of the output vector. A smaller language model outputs a smaller token vector that stores less information but can process a document significantly faster. See Figure 4 and Table 1 for a quantitative evaluation of language models of different sizes.

Classifiers

After obtaining the token representations using the language model, the text is fed into the classifier network as an ordered list of numeric vectors, one for each token, which is then subsequently mapped onto corresponding probabilities

[9]"Mr. Vogel (bird) is CEO of Test GmbH (equiv. LLC)."

[10]"The early bird catches the worm."

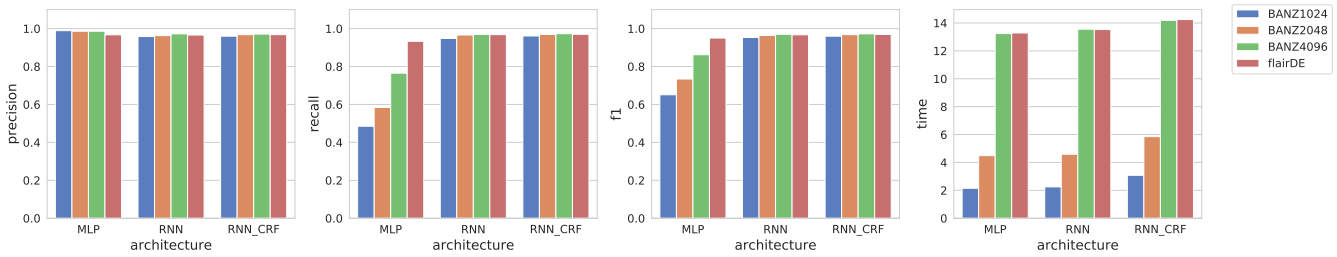

Figure 4: Influence of language model on precision, recall, F_1-score and inference time on evaluation documents. Precision and recall are reported without post-processing. Inference time measured in seconds per document (10 pages). We see that for the RNN based architectures, the choice of language model makes little difference in anonymization performance. However a smaller language model reduces the time it takes to process one document significantly. Note that there are no major differences in processing time between classifier architectures, the language model is the main contributor to processing time.

for each of the 7 named entities (*0*, *ORG*, *PER*, *LOC*, *PROD*, *SEG* and *OTH*). During training, the network is trained to predict the expert annotated labels for each token by minimizing the cross-entropy loss. Once the network is trained in this fashion, during inference, the label with the highest probability is predicted. We consider three different classifiers architectures:

MLP First, we consider a simple fully connected network (*multi-layer perceptron*) that takes each token representation individually, passes it through several layers and outputs probabilities for each of the 7 named entities. In this case, the prediction for each token is treated independently and relies solely on the contextual representation provided by the language model. This classifier is preferred because of faster inference time and easier interpretability of results.

RNN Although a simple MLP is sufficient to classify a token since the representation contains the context, it is still beneficial to process the text using a *recurrent neural net* which further enhances the context and more importantly the required span of context can be trained for the given task. For this reason, we consider a bi-directional variant of *Long Short Term Memory* (*LSTM*) [Hochreiter and Schmidhuber1997] which traverses the list of vectors in both directions, processing stored context information from previous tokens and the current token. The outputs along both directions (forward and backward) are concatenated and passed through a final fully connected prediction layer mapping to probabilities for each of the 7 named entities.

RNN + CRF With MLP and RNN, the prediction of each token is treated independently. In order to incorporate dependencies between predicted labels, the fully connected layer from output states of the RNN to the output layer can be replaced by a *conditional random field* (*CRF*) [Lafferty, McCallum, and Pereira2001] that learns a mapping of sequences of representations taking into account the predicted labels of consecutive tokens.

Post-processing

As discussed in the section , we also provide an option in our application to anonymize URLs, dates, numbers and e-mail addresses. Since they mostly have regular patterns, we have implemented regular expressions to detect these entities. Also, there might be tokens in the given text which are predicted as sensitive in one place and as not sensitive in other places. We intend to avoid this behavior and have higher preference to recall than precision for our anonymization. To this end, we propose to apply a post-processing step that ensures consistency in the predicted labels. For instance, a token (e.g. a persons name) that is predicted as a named entity once in the document is always replaced by the corresponding label, even if the classifier predicted it as non-sensitive in another sentence.

Experiments and Results

Datasets and Models

In the following subsections we describe the specific datasets, architectures and techniques used for training language models and classifiers.

Language Model Corpus As discussed in the previous section, in order to obtain contextual representations for tokens, we consider different language models. The baseline model that we use is a pre-trained language model provided by the flair framework which is trained on a large general corpus of German senetences consisting of 500 million words. We refer the embedding obtained using this model as *flairDE*. The language corpus used in the training of this embedding might cause licensing issues, e.g. the Wikipedia corpus is distributed under *GNU Free Documentation License* and *Creative Commons Attribution-Share-Alike 3.0 License*, which prohibit commercial use without adapting the same license to the project. Additionally, a language model trained on data that is similar to the financial text might provide an advantage over a language model trained on general language data and a custom language model allows for tuning the embedding size in order to optimize run-time. We therefore train language models on a corpus of language data from *Bundesanzeiger*[11] (*BANZ*), consisting of 19000 german financial documents (200 million words).

[11] https://www.bundesanzeiger.de/ebanzwww/wexsservlet

		On financial documents						On Germeval		
		Before post-processing			After post-processing			Before post-processing		
Architecture	Embedding	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
MLP	BANZ1024	**0.989**	0.485	0.651	**0.960**	0.682	0.797	0.928	0.076	0.140
MLP	BANZ2048	0.986	0.584	0.734	0.954	0.777	0.856	**0.938**	0.136	0.238
MLP	BANZ4096	0.986	0.765	0.862	0.938	0.867	0.901	0.923	0.179	0.300
MLP	flairDE	0.967	0.933	0.950	0.905	0.968	0.935	0.669	0.793	0.726
RNN	BANZ1024	0.958	0.948	0.953	0.897	0.976	0.935	0.720	0.646	0.681
RNN	BANZ2048	0.963	0.966	0.964	0.907	0.985	0.944	0.778	0.622	0.691
RNN	BANZ4096	0.972	0.969	0.970	0.915	0.986	**0.949**	0.815	0.638	0.716
RNN	flairDE	0.966	0.968	0.967	0.906	0.988	0.945	0.808	**0.848**	0.828
RNN_CRF	BANZ1024	0.960	0.961	0.960	0.897	0.983	0.938	0.741	0.684	0.711
RNN_CRF	BANZ2048	0.968	0.969	0.968	0.910	0.987	0.947	0.784	0.654	0.713
RNN_CRF	BANZ4096	0.971	**0.973**	**0.972**	0.910	0.987	0.947	0.796	0.675	0.731
RNN_CRF	flairDE	0.968	0.970	0.969	0.903	**0.990**	0.945	0.824	0.840	**0.832**
flairNER	flairDE	0.938	0.779	0.851	0.853	0.897	0.874	0.889	0.755	0.817

Table 1: Quantitative evaluation of all described language models and classifiers on the NER evaluation dataset of financial documents and the GermEval dataset. We provide all metrics on the positive class (*PER*, *ORG* and *LOC*). The best performance for each metric is marked bold for each column respectively. Post-processing for these classes only consists of ensuring label consistency. We do not evaluate post-processing for Germeval since its structure (independent sentences) does not fit our post-processing methods.

Document corpus We train our deep learning classifier models using a corpus of 407 published German financial documents, annotated manually by domain experts. We split the dataset into 305 training and 102 validation documents. Once a model is trained, we provide a final evaluation dataset consisting of 45 thoroughly annotated documents. This evaluation dataset contains a total of 189k tokens, 17k (9.1%) of which belong to one of the classes *ORG*, *LOC* and *PER*. In order to provide results comparable to other NER and anonymization projects, we additionally evaluate all trained models on the *GermEval 2014 NER Shared Task* corpus [Benikova et al.2014], consisting of 29k sentences annotated for NER with a total of approximately 590k tokens, 8.4% of which are named entities.

Language Models To train and use a language model on our data, we employ the framework provided by the *flair* python-package[12]. It implements a bidirectional LSTM on a character-level. We train language models on the *BANZ*-corpus with 1024, 2048 and 4096 dimensions. These are denoted by *BANZ1024*, *BANZ2048*, *BANZ4096* respectively. We train for 100 epochs using the default parameters suggested by the package.

Classifiers The RNN classifier as suggested by [Akbik, Blythe, and Vollgraf2018] is a one-layer BiLSTM with a hidden representation of 256 dimensions. We use the framework provided by the *flair* package to train RNN-based NER classifiers on the NER training dataset. We train for 100 epochs using the default parameters suggested by the package. Each MLP model consists of one intermediate hidden layer, mapping the input onto a lower dimensional representation. This hidden representation is then mapped onto the 7 dimensional output vector. The number of neurons in the intermediate hidden layer are 500, 500, 1000 depending on the input dimension 1024, 2048 and 4096 respectively. We train the MLP classifier for 100 epochs, using a batch size of 100 tokens. As optimizer, we use Adadelta with a learning rate of 0.1 and weight decay of 1e-5. Further, we also consider a pre-trained NER model provided by the *flair* package, which is a RNN+CRF classifier trained on the CoNLL-2003 German NER dataset [Tjong Kim Sang and De Meulder2003] using a general corpus language model. We denote this classifier as *flairNER* and provide evaluations as a baseline in the evaluation.

[12]https://github.com/zalandoresearch/flair

Results and Discussion

In this section, we present quantitative results on the performance of the described language models and classifiers. For our task of anonymization, it is desired to have a good binary classification performance, i.e. we tolerate a *PER* entity being tagged as an *ORG* entity and at the same time, we consider a *PER* entity tagged as *0* as a mis-classification and vice versa. For this reason, before evaluation all predicted and annotated tags are re-mapped onto two classes only, the negative class *0* indicating they are not sensitive entities and the positive class *1* indicating such tokens to be anonymized. Further, we are mostly interested in the performance on the positive class and therefore provide its metrics (precision, recall and F_1-score) only. Due to the lack of reliable available data for *SEG*, *PROD* and *OTH*, we do not consider them during this evaluation.

Table 1 presents the complete experimental results with different classifier architectures and language models. The evaluation on financial documents suggests that the RNN+CRF achieves the best performance, at over 97%

recall without post-processing and around 99% after post-processing, without compromising precision of over 90%. This results in a near complete anonymization of the entire document with very little unnecessarily anonymized words. Using domain-specific language model gives slight improvements over general language models for RNN-based classifiers. On the other-hand, the general corpus was beneficial while using a MLP classifier.

In order to evaluate the generalizability of our classifiers, we evaluate our models on GermEval dataset. For this evaluation, we do not apply any post-processing since it contains only sentences obtained from different sources and they do not follow any document structure. The results suggest that the RNN classifiers using a general language model performs better than one trained only on financial documents, which is expected sicne the sentences in GermEval corresponds to sentences from a variety of sources. Nevertheless, the performance is comparable to the current state-of-the-art for NER.

Further, the pre-trained NER classifier, trained on a general language German NER corpus, only gives a recall of 84%, 93% on the financial documents, without and with post-processing respectively.

Figure 4 captures the influence of language model on the performance metrics. From the run-time and recall plots, we can observe that even with the smaller domain-specific language models, the RNN classifiers are able to out-perform the general language model, while reducing the run times of the anonymization process by over 50%.

Conclusion

We presented a method to reliably anonymize the names of persons, locations and organisations using state-of-the-art deep learning techniques, as well as URLs, telephone numbers, dates and other numbers using classical rule-based approaches in financial documents. For internal use this method can be applied to a single document or entire document corpora using a web-based application and a python-based API. This allows for pre-processing of documents that can then be used by developers and researchers to train and evaluate further models for machine learning on financial data (e.g. [Sifa et al.2019]).

A quantitative evaluation of language models and text classifiers shows that domain-specific training of language models improve classification performance and smaller language models significantly improve runtime while maintaining anonymization performance. As future work, we would like to incorporate methods to anonymize additional identifying information (e.g. the segments the organisation operates in) as well as analyze the impact of anonymized data as inputs for the training of machine learning algorithms over the original text.

Acknowledgement

The authors of this work were supported in parts by the Fraunhofer Research Center for Machine Learning (RCML) within the Fraunhofer Cluster of Excellence Cognitive Internet Technologies (CCIT) and by the Competence Center for Machine Learning Rhine Ruhr (ML2R) which is funded by the Federal Ministry of Education and Research of Germany (grant no. 01—S18038A). We gratefully acknowledge this support.

References

Akbik, A.; Blythe, D.; and Vollgraf, R. 2018. Contextual String Embeddings for Sequence Labeling. In *Proc. of Int. Con. on Computational Linguistics*, 1638–1649.

Benikova, D.; Biemann, C.; Kisselew, M.; and Padó, S. 2014. GermEval Named Entity Recognition: Companion paper. *Proc. of the KONVENS GermEval Shared Task on Named Entity Recognition, Hildesheim, Germany* 104–112.

Ferrández, O.; South, B.; Shen, S.; et al. 2012. BoB, a Best-of-Breed Automated Text De-identification System for VHA Clinical Documents. *Journal of the American Medical Informatics Association* 20(1):77–83.

Gardner, J., and Xiong, L. 2008. HIDE: an Integrated System for Health Information De-Identification. In *Proc. on. International Symposium on Computer-Based Medical Systems*, 254–259.

Gola, P.; Eichler, C.; Franck, L.; et al. 2017. Datenschutzgrundverordnung: Ds-gvo. Art. 6, paragraph 255.

Hochreiter, S., and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural computation* 9:1735–80.

Lafferty, J. D.; McCallum, A.; and Pereira, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of Int. Conf. on Machine Learning*, ICML 01, 282289.

Li, J.; Sun, A.; Han, J.; and Li, C. 2018. A Survey on Deep Learning for Named Entity Recognition.

Mikolov, T.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781.

Neamatullah, I.; Douglass, M. M.; Li-wei, H. L.; et al. 2008. Automated de-identification of free-text medical records. *BMC Medical Informatics and Decision Making* 8(1):32.

Nguyen, N., and Guo, Y. 2007. Comparisons of Sequence Labeling Algorithms and Extensions. volume 227, 681–688.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. GloVe: Global Vectors for Word Representation. In *Proc. of Empirical Methods in Natural Language Processing*, 1532–1543.

Sifa, R.; Ladi, A.; Pielka, M.; and Ramamurthy, R. 2019. Towards Automated Auditing with Machine Learning. In *Proc. of Symposium on Document Engineering*.

Sweeney, L. 1996. Replacing Personally-Identifying Information in Medical Records, the Scrub System. In *Proc. of the AMIA Annual Fall Symposium*, 333.

Tjong Kim Sang, E. F., and De Meulder, F. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In Daelemans, W., and Osborne, M., eds., *Proc. of CoNLL*, 142–147.

Wellner, B.; Huyck, M.; Mardis, S.; et al. 2007. Rapidly Retargetable Approaches to De-Identification in Medical Records. *Journal of the American Medical Informatics Association* 14(5):564–573.