future internet

MDPI

Article

Generating Synthetic Training Data for Supervised De-Identification of Electronic Health Records

Claudia Alessandra Libbi [1,2], Jan Trienes [2,3,*], Dolf Trieschnigg [2] and Christin Seifert [1,3]

[1] Faculty of EEMCS, University of Twente, PO Box 217, 7500 AE Enschede, The Netherlands; alelib29@gmail.com (C.A.L.); christin.seifert@uni-due.de (C.S.)
[2] Nedap Healthcare, 7141 DC Groenlo, The Netherlands; dolf.trieschnigg@nedap.com
[3] Institute for Artificial Intelligence in Medicine, University of Duisburg-Essen, 45131 Essen, Germany
* Correspondence: jan.trienes@uni-due.de

Abstract: A major hurdle in the development of natural language processing (NLP) methods for Electronic Health Records (EHRs) is the lack of large, annotated datasets. Privacy concerns prevent the distribution of EHRs, and the annotation of data is known to be costly and cumbersome. Synthetic data presents a promising solution to the privacy concern, if synthetic data has comparable utility to real data and if it preserves the privacy of patients. However, the generation of synthetic text alone is not useful for NLP because of the lack of annotations. In this work, we propose the use of neural language models (LSTM and GPT-2) for generating artificial EHR text jointly with annotations for named-entity recognition. Our experiments show that artificial documents can be used to train a supervised named-entity recognition model for de-identification, which outperforms a state-of-the-art rule-based baseline. Moreover, we show that combining real data with synthetic data improves the recall of the method, without manual annotation effort. We conduct a user study to gain insights on the privacy of artificial text. We highlight privacy risks associated with language models to inform future research on privacy-preserving automated text generation and metrics for evaluating privacy-preservation during text generation.

Keywords: natural language processing; medical records; privacy protection; synthetic text; generative language models; named-entity recognition; natural language generation

Citation: Libbi, C.A.; Trienes, J.; Trieschnigg, D.; Seifert, C. Generating Synthetic Training Data for Supervised De-Identification of electronic health records. *Future Internet* **2021**, *13*, 136. https://doi.org/10.3390/fi13050136

Academic Editor: Marco Pota

Received: 26 April 2021
Accepted: 17 May 2021
Published: 20 May 2021

1. Introduction

Narrative text in electronic health records (EHRs) is a rich resource to advance medical and machine learning research. To make this unstructured information suitable for clinical applications, there is a large demand for natural language processing (NLP) solutions that extract clinically relevant information from the raw text [1]. A major hurdle in the development of NLP models for healthcare is the lack of large, annotated training data. There are two reasons for this. First, privacy concerns prevent sharing of clinical data with other researchers. Second, annotating data is a cumbersome and costly process which is impractical for many organizations, especially at the scale demanded by modern NLP models.

Synthetic data has been proposed as a promising alternative to real data. It addresses the privacy concern simply by not describing real persons [2]. Furthermore, if task-relevant properties of the real data are maintained in the synthetic data, it is also of comparable utility [2]. We envision that researchers use synthetic data to work on shared tasks where real data cannot be shared because of privacy concerns. In addition, even within the bounds of a research institute, real data may have certain access restrictions. Using synthetic data as a surrogate for the real data can help organizations to comply with privacy regulations. Besides addressing the privacy concerns, synthetic data is an effective way to increase the amount of available data without additional costs because of its additive nature [3,4]. Prior

work showed exciting results when generating both structured [5] and unstructured medical data [2]. In particular, recent advances in neural language modeling show promising results in generating high-quality and realistic text [6].

However, the generation of synthetic text alone does not make it useful for training of NLP models because of the lack of annotations. In this paper, we propose the use of language models to jointly generate synthetic text and training annotations for named-entity recognition (NER) methods. Our idea is to add in-text annotations to the language model training data in form of special tokens to delimit start/end boundaries of named entities (Figure 1). The source of those in-text annotations can be a (potentially noisy) pre-trained model or manual annotation. By adding the special tokens to the training data, they explicitly become part of the language modeling objective. In that way, language models learn to produce text that is automatically annotated for downstream NER tasks—we refer to them as "structure-aware language models." Below, we will briefly outline our research pipeline; see Figure 2 for an overview.

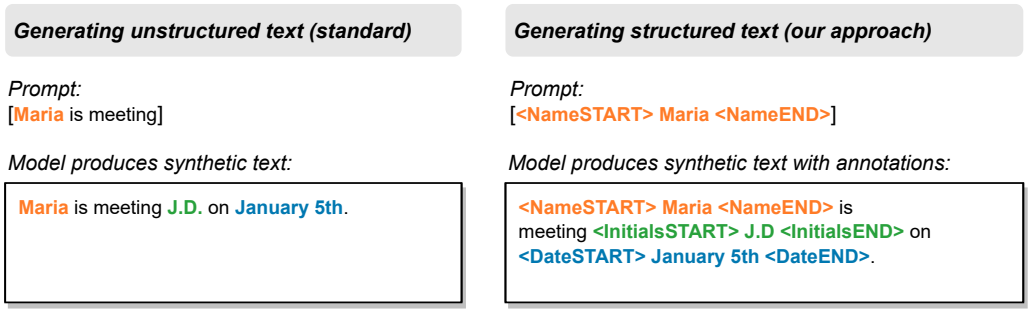

Figure 1. Illustrative example comparing standard text generation with the approach taken in this paper. We introduce special tokens to delimit protected health information (PHI). These tokens can be learned and generated like any other token by the language models. A prompt of three tokens defines the initial context.

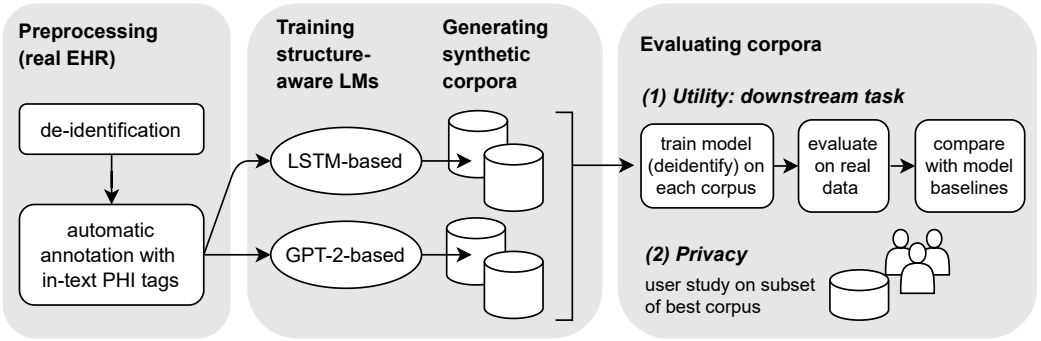

Figure 2. Overview of this study. (1) Raw, EHR text is automatically de-identified and annotated with in-text PHI labels. (2) Pre-processed text is used to train two "structure-aware" language models: an LSTM and GPT-2. (3) Using different decoding strategies, two synthetic corpora are generated from each language model. (4) Synthetic text is evaluated regarding utility and privacy. (4.1) Utility is measured by comparing the performance of machine learning models trained on synthetic data with models trained on real data. (4.2) For the privacy evaluation, ROUGE n-gram overlap and retrieval-based BM25 scoring is used to select the most similar real documents. Afterwards, the synthetic-real document pairs are presented to participants in a user study.

We compare two state-of-the-art language modeling approaches for the generation of synthetic EHR notes: a Long Short-Term Memory (LSTM) network [7] and a transformer-based network (GPT-2) [8]. To train these language models, we use a large and heterogeneous corpus of one million Dutch EHR notes. This dataset is unique in that it entails records of multiple institutions and care domains in the Netherlands.

We evaluate our approach by considering both utility and privacy of synthetic text. For utility, we choose the challenging NLP downstream task of de-identification. The objective of de-identification is to detect instances of protected health information (PHI) in text, such as names, dates, addresses and professions [9]. After detection, the PHI is masked or removed for privacy protection. De-identification as a downstream task is particularly interesting, because it requires sensitive data which would not be shared otherwise. We consider utility of synthetic data under two use-cases: (1) as a replacement for real data (e.g., in data sharing), and (2) as a data augmentation method to extend a (possibly small) set of real documents. To add in-text annotations for the de-identification downstream task, we obtain heuristic PHI annotations on the language model training data through a pre-trained de-identification method called "deidentify" [10]. Note that this setup is not limited to de-identification. In principle, any other information extraction method (or manual annotation) could act as a source for initial training annotations.

To evaluate privacy of synthetic records, we design a user study where participants are presented with the synthetic documents that entail the highest risks of privacy disclosure. As we have no 1-to-1 correspondence between real and synthetic documents, we devise a method to collect high-risk candidates for evaluation. We posit that synthetic documents with a high similarity to real documents have a higher risk of disclosing privacy sensitive information. We use ROUGE n-gram overlap [11] and retrieval-based BM25 scoring [12] to collect the set of candidate documents. Participants were asked to make judgments on the existence and replication of sensitive data in those examples with the goal to (1) evaluate the privacy of our synthetic data, and (2) to inform and motivate future research and privacy policies on the privacy risk assessment of free text that looks beyond PHI.

This paper makes the following contributions:

- We show that neural language models can be used successfully to generate artificial text with in-line annotations. Despite varying syntactic and stylistic properties, as well as topical incoherence, they are of sufficient utility to be used for training downstream machine learning models.
- Our user study provides insights into potential privacy threats associated with generative language models for synthetic EHR notes. These directly inform research on the development of automatic privacy evaluations for natural language.

We release the code of this study at: https://github.com/nedap/mdpi2021-textgen, accessed on 17 May 2021.

2. Background and Related Work

In this section, we provide a summary of related work on the generation of synthetic EHRs (Section 2.1), as well as the evaluation of privacy (Section 2.2). Furthermore, we give general background on language modeling and decoding methods (Section 2.3).

2.1. Generating Synthetic EHR Notes

The generation of synthetic EHR text for use in medical NLP is still at an early stage [3]. Most studies focus on the creation of English EHR text, using hospital discharge summaries from the MIMIC-III database [7,8,13,14]. In addition, a corpus of English Mental Health Records was explored [15]. Unlike the mixed healthcare data used in this study, these EHR notes have a more consistent, template-like structure and contain medical jargon, lending itself to clinical/biomedical downstream tasks found in related work [8,13–15]. Most of these studies focused on classification downstream tasks. To the best of our knowledge, we are the first study that attempts to generate synthetic data for sequence labeling (NER).

Decoding from language models is the predominant approach in related work to generate synthetic text. Approaches include unigram-language models and LSTMs [7], as well as transformer-based methods such as GPT-2 [13–15]. In particular, Amin-Nejad et al. [8] concluded that GPT-2 was suitable for text generation in a low-resource scenario. In this research, we compare a standard LSTM-based model with a transformer-based model (GPT-2). At the time this research was conducted, the only pre-trained Dutch transformer

models available were BERT-based [16,17]. Since no pre-trained Dutch GPT-2 model existed, we chose to fine-tune an openly available English GPT-2 [6] on our data for this purpose.

Prior studies also consider different ways to generate EHR notes with a pre-defined topic. These approaches include conditional generation on clinical context [8,13] and guiding by keyphrases extracted from an original note [14,15,18]. As a result, the synthetic notes inherently have one-to-one relations with the original data. In this study, we do not use the conditional text generation approaches for two reasons. First, the NER use-case does not require strong guarantees on the topic of synthetic training examples. This is different from downstream tasks like classification. Second, we do not want that synthetic notes have a one-to-one link to real data. We assume that this benefits privacy protection. Instead of the conditional generation mentioned above, we use short prompts to generate whole EHR notes without a pre-defined topic.

2.2. Evaluating Privacy of Synthetic EHR Notes

While privacy preservation is one of the main motivations for the generation of synthetic EHR, related research did not always report privacy of generated corpora or propose methods for the evaluation. For example, Amin-Nejad et al. [8] and Liu [13] used similarity metrics as intrinsic measure to compare real and synthetic notes, but did not draw further conclusions on privacy. Melamud and Shivade [7] propose an empirical measure to quantify the risk of information leakage based on differential privacy. However, the calculation of this measure requires training a prohibitively large amount of models and does not directly provide information on the privacy of the generated data itself. Embedding differential privacy in the model training process, would theoretically ensure privacy [19]. However, the known trade-off between privacy and utility [7,19] dissuaded us from training differentially private models, as the primary focus was on achieving high utility. To draw conclusions about the privacy of our synthetic records, we develop a simple method to query "high-risk" candidates from the synthetic documents based on shallow text similarity metrics. We conduct a user study to investigate potential privacy issues concerning these records.

2.3. Background on Natural Language Generation

In the area of natural language generation (NLG) there are several approaches to generate artificial text. In this study, two neural methods with different architectures are considered, both of which are based on training a language model on text with the desired features (i.e., the one that we want to model). LSTM models are recurrent neural networks that process input sequentially and are able to learn long-term dependencies [20]. They are now widely used in natural language generation. More recently, Vaswani et al. [21] introduced the transformer architecture, which does not represent text sequentially, but can attend to the whole input in parallel and therefore store syntactic and semantic information on a higher level [6,21]. "GPT-2" or the "Generative Pre-Trained Transformer (2)" is an open-source, transformer-based language model by OpenAI [6], which was trained on 40 GB of text crawled from the internet. While already capable as a general-purpose model for English text [6], fine-tuning (i.e., transfer learning) can be used to learn a domain-specific language (e.g., non-English, medical jargon, writing style) while still taking advantage of the existing learned language patterns [22,23].

To use a language model for text generation, several decoding algorithms exist to pick a sequence of tokens that is likely to exist, given the language model. Depending on the chosen algorithm, the potential differences in outcome can be summarized as: (1) diversity, i.e., how much variation there is in different outputs, given the same input prompt, and (2) quality of the generated text, which may include how quickly it degrades with text length, and how meaningful, specific and repetitive it is [4,24–26]. As opposed to tasks like machine-translation (the output sequence must be consistent with the input sequence), open-ended language generation tasks demand higher diversity and creativity of output. Most commonly used are maximization-based decoding strategies (e.g., beam search).

However, these greedy methods tend to produce repetitive outputs. Sampling-based methods like temperature sampling and nucleus sampling generate more varied text [24].

3. Materials and Methods

This section describes our experimental setup including the dataset, procedure for training the language models and evaluation of utility and privacy.

3.1. Corpus for Language Modeling

To construct a large and heterogeneous dataset for language model training, we sample documents from the EHRs of 39 healthcare organizations in the Netherlands. Three domains of healthcare are represented within this sample: elderly care, mental care and disabled care. All text was written by trained care professionals such as nurses and general practitioners, and the language of reporting is Dutch. A wide variety of document types is present in this sample. This includes intake forms, progress notes, communications between care givers, and medical measurements. While some documents follow domain-specific conventions, the length, writing style and structure differs substantially across reports. The sample consists of 1.06 million reports with approximately 52 million tokens and a vocabulary size of 335 thousand. For language model training, we randomly split the dataset into training, validation, and testing sets with a 80/10/10 ratio. We received approval for the collection and use of the dataset from the privacy board of Nedap Healthcare.

3.2. Pre-Processing and Automatically Annotating the Language Modeling Data

Before using the collected real data for developing the language model, we pseudonymize it as follows. First, we detect PHI using a pre-trained de-identification tool for Dutch healthcare records called "deidentify" [10]. The "deidentify" model is a BiLSTM-CRF trained on Dutch healthcare records in the domains of elderly care, mental care and disabled care. The data is highly similar to the data used in this study and we expect comparable effectiveness to the results reported in the original paper (entity-level F1 of 0.893 [10]). After de-identfication, we replace the PHI with random, but realistic surrogates [27]. The surrogate PHI will serve as "ground-truth" annotations in the downstream NLP task (Section 3.4). Table 1 shows the distribution of PHI in the language modeling corpus. To make annotations explicitly part of the language modeling objective, we add in-text annotations from the PHI offsets (as shown in Figure 1). Each annotation is delimited by a special `<xSTART>` and `<xEND>` token where `x` stands for the entity type. We acknowledge that the automatically annotated PHI will be noisy. However, we assume that quality is sufficient for an initial exploration of the viability of our synthetic data generation approach. Unless otherwise stated, we use the spaCy (https://github.com/explosion/spaCy, accessed on 19 May 2021) tokenizer and replace newlines with a `<PAR>` token.

We would like to highlight the motivation for annotating the real documents (i.e., before language modeling) and not the synthetic documents (i.e., after language generation). In theory, because we have a pre-trained NER model available, both options are possible. However, there are two reasons why we propose to make the annotations part of the language modeling. First, the language models may learn to generate novel entities that a pre-trained model would not detect (we provide tentative evidence for this in Section 4.2.2). Second, because we could generate synthetic datasets many orders of magnitude larger than the source data, it is more efficient to annotate the language modeling data. The second argument especially holds if no pre-trained annotation model is available and records have to be manually annotated.

Table 1. Distribution of PHI tags in the 52 million token corpus used to develop the language models (i.e., real data). PHI was tagged by an automatic de-identification routine.

PHI Tag	Count	% of Total
Name	782,499	59.74
Date	202,929	15.49
Initials	181,811	13.88
Address	46,387	3.54
Care Institute	38,669	2.95
Organization	37,284	2.85
Internal Location	6977	0.53
Phone/Fax	3843	0.29
Age	3350	0.26
Email	2539	0.19
Hospital	2425	0.19
Profession	537	0.04
URL/IP	326	0.02
ID	232	0.02
Other	105	0.01
SSN	6	0.00
Total	1,309,919	100

3.3. Generative Language Models

We compare two language modeling approaches for the generation of synthetic corpora: LSTM-based [20] and transformer-based (GPT-2) [6]. Below, we outline the model architectures as well as the decoding methods to generate four synthetic corpora. For a summary, see Tables 2 and 3.

3.3.1. LSTM-Based Model

Because of their success in generating English EHR, we re-implement the method including hyperparameters by Melamud and Shivade [7]. The model is a 2-layer LSTM with 650 hidden-units, an embedding layer of size 650 and a softmax output layer. Input and output weights are tied. The model is trained for 50 epochs using vanilla gradient descent, a batch size of 20 and a sequence length of 35. We also use learning rate back-off from [7]. The initial learning rate is set to 20 and reduced by a factor of 4 after every epoch where the validation loss did not decrease. The minimum learning rate is set to 0.1. For efficiency reasons, we replace tokens that occur fewer than 10 times in the training data with `<unk>` [7].

3.3.2. Transformer-Based Model (GPT-2)

From the family of transformer models, we use GPT-2 [6]. Prior work showed promising results using GPT-2 for the generation of English EHR [8]. To the best of our knowledge, there is no Dutch GPT-2 model for the clinical domain which we could re-use. However, prior work showed that pre-trained English models can be adapted to the Dutch language with smaller computational demand than training from scratch [28]. The intuition is, that the Dutch and English language share similar language rules and even (sub-)words. Below, we provide a summary of this fine-tuning process.

Adapting the vocabulary: We train a byte-pair-encoding (BPE) tokenizer on our Dutch EHR corpus. All sub-word embeddings are randomly initialized. To benefit from the pre-trained English GPT-2 model (small variant) [6], we copy embeddings that are shared between the English and Dutch tokenizer. To account for the in-text annotations, we add a tokenization rule to not split PHI tags into sub-words.

Fine-tuning the model: The layers of the pre-trained GPT-2 model represent text at different abstraction levels. For transfer learning, the key is to take advantage of the previously learned information that is relevant for the current task, but adjust representations such that they are suitable for the new language and domain-specific terminology. To do

so, layers are split into groups and we use gradual unfreezing with differential learning rates, such that the last layer group (with corpus-specific information) is changed more than the first ones, where learned representations can be re-used. To train layer groups on our data, we used the one-cycle-policy [29], where learning rates are scheduled with cosine annealing. Our GPT-2 model was split into four layer groups which were trained in 5 epochs. We provide additional details on model and fine-tuning steps in Table 2 and Appendix A.

Table 2. Summary of language models used to generate synthetic text. Note that the test perplexity cannot be directly compared due to the difference in vocabulary.

	LSTM	GPT2
Tokenizer	spaCy, replace low-frequency tokens (<= 10) with <unk>	Trained English "ByteLevelBPE Tokenizer" on Dutch corpus, while keeping embeddings for common tokens.
Model	2-layer LSTM (650 input embedding size, 650 hidden units, softmax output) [7]	GPT-2 English small (12-layer, 768-hidden, 12-heads, 117M parameters before fine-tuning) [6]
Vocabulary	49,978 tokens	50,257 tokens
Parameters	39,307,380	163,037,184 (after fine-tuning)
Perplexity	32.1	38.8

3.3.3. Decoding Methods for Generation of Synthetic Corpora

Using the LSTM, GPT-2 and different decoding methods, we generated four synthetic corpora of approximately 1 million tokens each (Table 3). As initial context for each report, we selected random prompts of length 3. These were sampled from held-out EHRs to minimize the possibility of reconstructing real documents during generation. Generation of a text was terminated either when a maximum token count was reached, or when the model produced an end-of-document token. For all corpora, we impose a subjective minimum document length of 50 tokens.

Following Holtzman et al. [24], we generate two corpora with nucleus sampling ($p = 0.95$, LSTM-p and GPT-p). Additionally, we implement the decoding methods of the papers that proposed the LSTM [7] and GPT-2 [8] for the generation of EHRs. For the LSTM, we generate a corpus with temperature sampling ($t = 1$, LSTM-temp). For the GPT-2 we use beam search ($n = 5$, GPT-beam) and exclude texts without PHI tags, as the corpus already had a lower overall number of tags which are essential for the utility in the downstream task. For both GPT-2 corpora, we set the generator to not repeat n-grams longer than 2 words within one text to increase variability. In rare cases, the language models produced annotations with trailing start/end tags. These malformed annotations were removed in an automatic post-processing step. We quantify how many annotations were removed in Section 4.1.1.

Table 3. Overview of language model decoding parameters to generate four synthetic corpora.

Corpus	Model	Generation Method	Tokens/Doc.
LSTM-p	LSTM	p-sampling ($p = 0.95$)	50–400
LSTM-temp	LSTM	Temperature sampling ($t = 1$)	50–500
GPT-p	GPT-2	p-sampling ($p = 0.95$)	50–400
GPT-beam	GPT-2	Beam search (beams $n = 5$)	50–500

3.4. Extrinsic Evaluation on NLP Downstream Task

To understand if the synthetic data and annotations have sufficient utility to be used for training of NLP models, we measure effectiveness in a de-identification downstream task. The objective of de-identification is to detect instances of PHI in text, such as names, dates, addresses and professions [9]. Ideally, a de-identification model trained on synthetic

data performs as good or better than a model trained on real data. To evaluate this, we train a BiLSTM-CRF de-identification model in three settings: (1) using real data, (2) extending real with synthetic data, and (3) using only synthetic data (Figure 3). As implementation for the BiLSTM-CRF, we use "deidentify" (https://github.com/nedap/deidentify, accessed on 19 May 2021) with the same architecture and hyperparameters as reported in the original paper [10]. As real data, we use the NUT corpus of that study with the same test split such that results are comparable. NUT consists of 1260 records with gold-standard PHI annotations.

The effectiveness of the de-identification models is measured by entity-level precision, recall and F1. The BiLSTM-CRF trained on real data is considered as the upper baseline for this problem. We also report scores of a rule-based system (DEDUCE [30]) which gives a performance estimate in the absence of any real or synthetic training data.

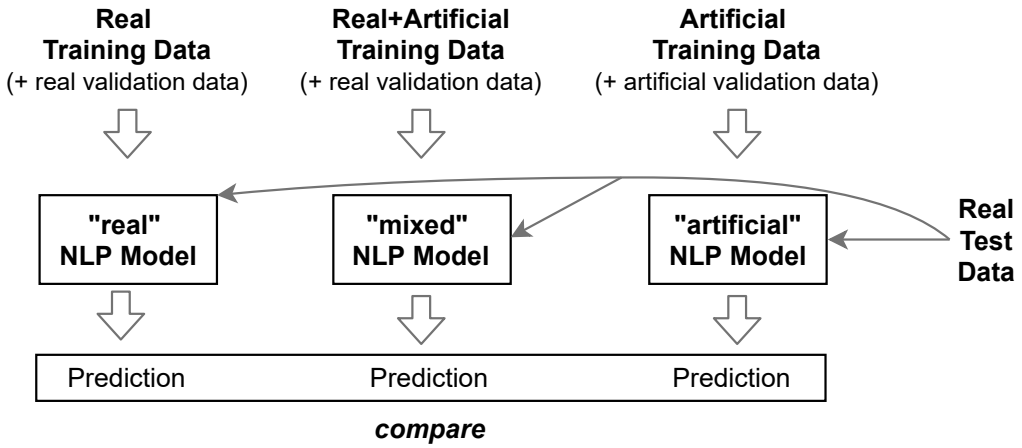

Figure 3. Overview of extrinsic evaluation procedure. We compare three settings: (1) a model trained on real data (baseline), (2) a "mixed" case, where we extend real data with synthetic data, and (3) only synthetic training data. All models were tested on real data (gold annotations). This evaluation setup extends Ive et al. [15] by step (2).

3.5. Privacy Evaluation

To gain insights into the privacy of synthetic data, we conducted a user study for a subset of synthetic documents from the corpus with highest utility in the downstream task. Our goal was to check whether any information "leaked" from the real data into the synthetic data, and whether this information could be used to re-identify an individual.

Finding potential worst cases for privacy. The assumption is that a privacy leak may have occurred when certain information of a real document reappears in a synthetic document. Similarly to the study by Choi et al. [31], we have no 1-to-1 correspondence between real and synthetic records. Let $s \in S$ be a synthetic document and $r \in R$ be a real document. Assuming that the likelihood of a privacy leak is higher when the proximity between s and r is high, we get a set of document pairs (SR) where for each s the most similar document r is retrieved as candidate source document (cf. Figure 4). We use three measures to obtain the most similar documents to a synthetic document: ROUGE-N recall [11], with $n = 3$ and with $n = 5$, and retrieval-based BM25 scoring [12]. We use standard BM25 parameters $b = 0.75$ and $k = 1.2$ [12].

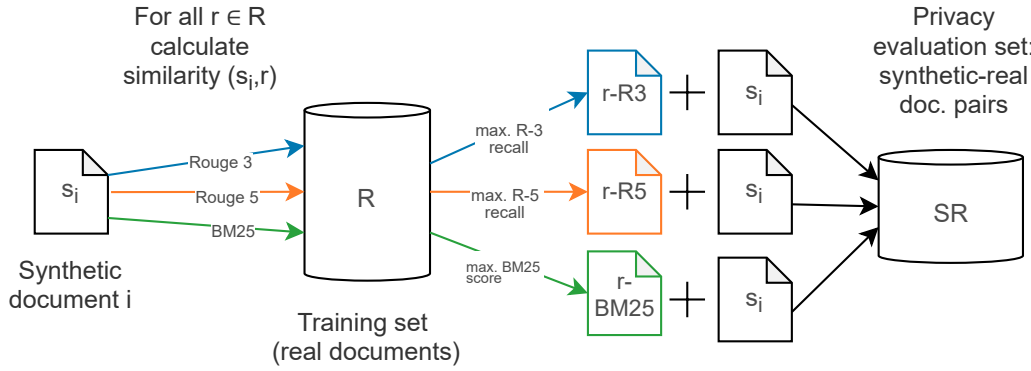

Figure 4. Illustration of method used to compile a set of similar synthetic-real document pairs for the privacy evaluation. For each synthetic document, we retrieve the most similar source documents from the real data, based on ROUGE n-gram overlap and BM25. The set *SR* contains the pooled result of this matching process, such that each synthetic document appears in three separate pairings: once with the top ROUGE-3 match, once with the top ROUGE-5 match and once with the top BM25 match.

Instead of randomly sampling synthetic documents for manual inspection, we used several filtering steps to maximize the probability of showing pairs with high similarity and readability during evaluation: We first sorted the documents by highest ROUGE scores. Afterwards, we removed duplicates, documents longer than 1000 characters (to control the reading effort of participants), and documents that received high similarity scores mostly based on structural elements (e.g., `<PAR>` tokens). We took the top 122 documents with highest ROUGE score for the user study. Full details of the filtering procedure are provided in Appendix D.

Participants were asked to answer the following questions for each pair of real/synthetic documents:

Q1: "Do you think the real doc provides enough information to identify a person?"
Q2: "Do you think the synthetic doc contains person identifying information?"
Q3: "Do you think that there is a link between the synthetic and real doc in the sense that it may identify someone in the real doc?"
Q4: "Please motivate your answer for Q3."

Questions 1–3 are on a 5-point Likert scale (Yes, Probably yes, Not sure, Probably not, No), and Q4 is an open text answer. Participants received a short introduction about the task and privacy. We supplied two trial documents for participants to get used to the task. These documents were excluded from analysis. The full questionnaire and participation instructions are given in Appendix D.

As the privacy sensitive data could not be shared with external parties, we recruited 12 participants from our institution (Nedap Healthcare). Due to the participant pool, there is a potential bias for technical and care related experts. We consider the impact for a privacy evaluation low, and indeed, because of their domain knowledge, participants have provided some helpful domain-related qualitative feedback. All participants were native Dutch speakers and each document pair was independently examined by two participants. We computed inter-participant agreement for each question with Cohen's Kappa. As the Likert scales produce ordinal data and there is a natural and relevant rank-order, we also calculated the Spearman's Rank-Order Correlation, to better capture the difference in participants disagreeing by, for example, answering "Yes" and "Probably" versus "Yes" and "No." This is especially relevant for the questions in this evaluation, which are hard to answer and likely to result in participants showing different levels of confidence due to personal differences. Both Kappa score and Spearman correlation were calculated per question, micro-averaged over all document pairs.

4. Results

In this section, we provide a quantitative and qualitative analysis of the generated synthetic data (Section 4.1). Afterwards, we discuss the utility of these data in the de-identification downstream task (Section 4.2). We conclude with the results of our user study on the privacy of synthetic documents (Section 4.3).

4.1. Does the Synthetic Data Resemble the Properties of Real Data?

For an ideal data generation method, we would expect that the synthesized data closely follows the characteristics of real data. We examine key summary statistics for each synthetic corpus and give a real corpus as reference (Table 4).

We make two observations. First, the synthetic corpora differ substantially in variety as quantified by the vocabulary size. At the extremes, the vocabulary of LSTM-temp is 3.7 times larger than the vocabulary of GPT-beam although they are comparable in size (approximately 1 million tokens). We expect that the variety has implications for the downstream utility of the datasets. Second, the GPT-2 p-sampling method generates sentences that are on average shorter than those of other methods. It is unclear what causes this specific behavior, but it indicates that the methods learn a different syntactic and stylistic representation of text. In summary, the synthetic text deviates from real text in key metrics. We investigate if it is still useful for downstream tasks in Section 4.2.

Table 4. Summary statistics of the synthetic corpora in reference to a real corpus (NUT).

	NUT [10]	**LSTM-p**	**LSTM-Temp**	**GPT-p**	**GPT-Beam**
Tokens	445,586	976,637	977,583	1,087,887	1,045,359
Vocabulary	30,252	23,052	29,485	12,149	8026
PHI instances	17,464	32,639	31,776	105,121	24,470
Sentences	43,682	70,527	72,140	128,773	83,634
Avg. tokens per sentence	10.2	13.8	13.6	8.4	12.5

4.1.1. Are the Synthetic PHI Annotations Well-Formed and Realistically Represented?

The syntactic quality of PHI annotations is good across all corpora. Between 97% and 99% of the annotations were well-formed (Table 5). We observe that the LSTM-based generators are slightly more consistent than the GPT-based generators. With respect to the distribution of PHI types, we observe that LSTM-based corpora stay closer to the real distribution (Figure 5). The GPT-2 model with beam-search decoder shows a pronounced bias for "Date" while the GPT-2 model with p-sampling boosts some of the rare PHI tags. Additionally, we note that the GPT-p corpus has substantially more PHI annotations (105 k) than the other corpora (24 k–33 k, Table 4). We analyze the impact of this in context of the downstream task (Section 4.2). A detailed report on the PHI frequencies per corpus can be found in Appendix B.

Table 5. A comparison of PHI tag consistency across synthetic corpora.

	LSTM-p	**LSTM-Temp**	**GPT-p**	**GPT-Beam**
Well-formed PHI tags	99.97%	99.89%	97.75%	98.84%
Malformed PHI tags	0.03%	0.11%	2.25%	1.16%

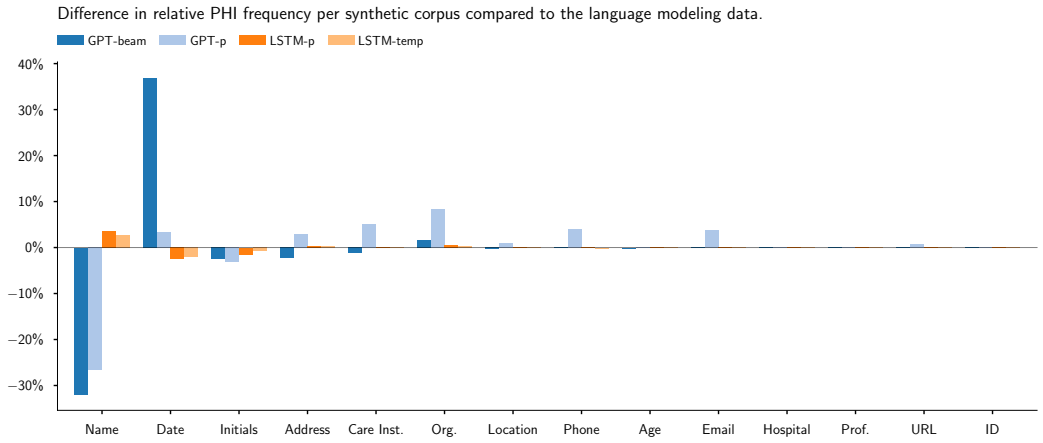

Figure 5. How well do the synthetic corpora reflect the real PHI distribution? This figure shows the differences to the PHI distribution of the language model training data (cf. Table 1).

4.1.2. Is the Generated Text Semantically Coherent?

To get a better understanding of the quality of generated text, we manually inspected random documents of the synthetic corpora (examples in Figure 6 and Appendix C). We make the following observations: while most texts are syntactically correct, the majority is incoherent. We hypothesize that the incoherence is caused by the large variety of reporting styles in the training corpus. This may have inhibited the language models to learn a specific type of text more accurately. Furthermore, we observe some replication of templates and phrases of real documents. An example of this is shown in Figure 6. This was most evident for texts generated by the GPT-2 with beam search. We give additional examples in Appendix C where we used the same prompt to generate text with all four approaches. In those examples, the LSTM texts are more varied, but also less coherent compared to the GPT-2 texts. Most notably, as the text length increases, the LSTM tends to deviate from the original context of the prompt while the GPT-2 stays closer to the topic.

<NameSTART> J. Smith **<NameEND>** did a check. Dental hygiene is good and the dentures are clean. No abnormalities of the mucous membranes.

Which instruction did you give: to the nursing staff on the ward

Specifics and poss. action (s): check oral hygiene. Brush the dentures with water and soap. Please sleep without dentures and store dry. In case of no improvement, consult the nursing staff. Take care when brushing the dentures: be careful with oral care!

To whom have you instructed: (incl. names of the nurses) caregivers

Follow up action
Prevention ass. **<NameSTART>** A. Baker **<NameEND>**
Prevention ass **<NameSTART>** E. Williams **<NameEND>** oral care

Action ass. ass. from the department of the dental care **<Care_InstituteSTART>** The Care Home **<Care_InstituteEND>** for the dry mouth and the mouth of mister **<NameSTART>** D. Johnson **<NameEND>** , **<Phone_faxSTART>** 89-1234567 **<Phone_faxEND>**

Figure 6. Text sample from the GPT-beam corpus (translated from Dutch, PHI highlighted and replaced with random identifiers). The structure of the generated text resembles a template that nurses used in the real data to report on dental hygiene of a patient.

4.2. Extrinsic Evaluation: Is the Utility of Synthetic Data Sufficient for Downstream Use?

We discuss the utility of synthetic data by considering two use cases: (1) as a replacement for real data, when real data are unavailable or cannot be shared, and (2) as a special form of data augmentation to generate cheap additional training examples.

4.2.1. Using Synthetic Data as a Replacement for Real Data

We find that de-identification models trained on any of the four synthetic corpora are not as effective as the real-data baseline (Table 6). However, the results are promising. In particular, the synthetic models outperform the rule-based method DEDUCE [30] by a large margin because of a substantial increase in recall (56.4% vs. 77.3% for LSTM-temp). The rule-based method relies on domain knowledge rather than real training examples and is therefore an interesting reference when no real training data is available. Overall, we observe that the LSTM-corpora provide better utility compared to the GPT-2 corpora, both in precision and recall (Table 6). Note that this is despite our earlier finding that the LSTM-corpora are less coherent (Section 4.1.2). For a task like de-identification, it seems that syntactic correctness is more important than coherency.

We study the influence of different PHI distributions in synthetic data by measuring precision and recall on a PHI-level (Table 7). We find that the de-identification model trained on LSTM data performs well on tags that appear frequently in the real data (e.g., Name and Date). However, the coverage of infrequent tags is insufficient (e.g., phone/fax and email). In contrast, the model trained on GPT-2 data is slightly less effective on the majority of PHI tags, but has a greater coverage of tags. We attribute this behavior to the GPT-2 p-sampling decoder, which seemingly boosted some of the rare PHI tags as discussed in Section 4.1.1. Considering the low effectiveness for identity-revealing tags, training de-identification models only on synthetic data is not yet practical. This is due to the high recall requirement for this task.

Table 6. Summary of downstream task performance. We train on the generated synthetic data and evaluate on real data with gold-standard annotations (*NUT* dataset [10]). Statistically significant improvements toward the *NUT (BiLSTM-CRF)* baseline are marked with ▲, and ° depicts no significant difference. The test is a two-tailed approximate randomization ($p < 0.01$).

Split: Train/val/Test	Dataset	Precision	Recall	F1
-/-/real	NUT (rule-based) [30]	0.807	0.564	0.664
real/real/real	NUT (BiLSTM-CRF) [10]	**0.925**	0.867	0.895
Use case 1: synthetic data as a replacement for real data				
synth/synth/real	LSTM-p	0.835	0.784	0.809
synth/synth/real	LSTM-temp	0.857	0.773	0.813
synth/synth/real	GPT-p	0.776	0.700	0.736
synth/synth/real	GPT-beam	0.823	0.688	0.749
Use case 2: synthetic data as data augmentation method				
real+synth/real/real	NUT+LSTM-temp	0.919°	**0.883**▲	**0.901**°
real+synth/real/real	NUT+LSTM-p	0.916°	0.879▲	0.897°

Finally, recall from Section 3.3.3 that we set the size of the synthetic corpora to 1 million tokens for all corpora. To understand how this setting influences the effectiveness of the downstream model, we train de-identification models on subsets of the synthetic data (LSTM-p corpus). We find that the learning curve flattens when using around 70% of the training data. This indicates that generating more data will not necessarily increase effectiveness. See Appendix E for details on this experiment.

Table 7. Entity-level precision and recall per PHI category. Comparing the baseline (*NUT*) with two models trained and validated on pure synthetic data (LSTM-p vs. GPT-p), as well as the mixed variant (NUT+LSTM-p) where the training set is composed of NUT and LSTM-p, but the validation set is the same as the one used in the baseline (real data). Highlighted values (bold) show improvements over the *NUT* baseline.

PHI Tag	NUT		GPT-p		LSTM-p		NUT+LSTM-p	
	Prec.	**Rec.**	**Prec.**	**Rec.**	**Prec.**	**Rec.**	**Prec.**	**Rec.**
Name	0.967	0.951	0.810	0.875	0.897	0.945	0.960	**0.959**
Date	0.929	0.910	0.910	0.813	0.889	**0.913**	**0.932**	**0.920**
Initials	0.896	0.629	0.456	0.146	0.595	0.421	0.822	**0.674**
Address	0.888	0.814	0.460	0.654	0.716	0.680	**0.901**	**0.878**
Care Institute	0.742	0.681	0.321	0.116	0.414	0.245	0.705	**0.718**
Organization	0.743	0.596	0.159	0.052	0.340	0.257	0.717	0.559
Internal Location	0.784	0.527	0.273	0.055	0.188	0.055	0.757	0.509
Phone/Fax	1.000	1.000	1.000	0.563	0.000	0.000	0.941	1.000
Age	0.757	0.683	0.320	0.195	**0.786**	0.268	**0.758**	0.610
Email	0.909	1.000	**1.000**	1.000	0.000	0.000	0.833	1.000
Hospital	0.778	0.700	0.333	0.100	0.300	0.300	**0.857**	0.600
Profession	0.833	0.238	0.000	0.000	0.000	0.000	**0.923**	**0.286**
URL/IP	1.000	0.750	1.000	0.500	0.000	0.000	1.000	0.750
ID	0.714	0.400	0.500	0.080	0.000	0.000	**0.786**	**0.440**
Other	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

4.2.2. Using Synthetic Data as Data Augmentation Method

As data annotation for de-identification is an expensive process, we experiment with a dataset that combines a small set of real documents (NUT) with a large set of synthetic documents. In this case, we focus on the synthetic corpora that showed best extrinsic utility (LSTM-temp and LSTM-p). We find that the combined datasets result in models with statistically significant improvements in recall with only an insignificant decrease in precision (Table 6). This increase in recall indicates that the language model produced novel PHI that was absent from the real training documents (NUT). At an entity level, we also observe that almost all PHI classes benefit from additional training examples (Table 7). Note that this performance improvement was achieved without additional manual annotation effort. The absence of an even larger improvement may be caused by a saturation of the model with only real data. Indeed, Trienes et al. [10] reported F1-scores for varying training set sizes (given real data), which show that at 100% of the training set, the learning curve has flattened.

4.3. Privacy Findings: Was Sensitive Information Leaked into the Synthetic Records?

The goal of the privacy evaluation was to learn whether the synthetic corpus (in this case the one with the highest utility, LSTM-p) contains documents that could leak privacy sensitive information from the real data. We sampled the synthetic-real document pairs with highest similarity and conducted a user study to find out what is considered person identifying information and whether there are cases where privacy has been compromised in the synthetic corpus.

4.3.1. Similarity between Real and Synthetic Documents

To give a first indication of potential privacy leaks, we report summary statistics for the ROUGE-N recall between all pairs of real/synthetic documents (Table 8). On average, the low n-gram recall suggests that the synthetic data is substantially different from the real data. However, we also find "high-risk cases" with large n-gram overlap. In some rare cases, documents were reproduced exactly (maximum ROUGE-N recall of 1). We focus on the top 122 synthetic documents with highest risk in the user study.

Table 8. Summary statistics for ROUGE-N recall over all real/synthetic document pairs and over the filtered subset of "high-risk" documents presented to participants in the user study.

	Over All Real/Synthetic Pairs				**Over 122 "High-Risk" Pairs**			
	Avg.	**Median**	**Min.**	**Max.**	**Avg.**	**Median**	**Min.**	**Max.**
ROUGE-3 recall	0.075	0.067	0.018	1.000	0.280	0.217	0.145	1.000
ROUGE-5 recall	0.031	0.026	0.000	1.000	0.207	0.143	0.025	1.000

4.3.2. User Study

Question 1 (Information to Re-Identify a Person in Real Document)

There was a fair agreement between participants (Cohen's Kappa $\kappa = 0.279$). The Spearman's rank-order coefficient of $\rho = 0.488$ (with $p = 1.19 \times 10^{-8}$) suggests that there is a (monotonic) positive association between the ratings of both participants. In 53 of 122 cases (Figure 7), participants agreed that the real document did not provide enough information to identify a person. In cases where participants answered with either "Probably" or "Yes," text often contained specific diagnoses (e.g., decubitus) in conjunction with PHI. Other examples were documents with specific psychological examination results (e.g., on personality, existence of suicidal thoughts, cognition, affect) or detailed descriptions of rare events (e.g., a person leaving a care home, an individual running away, descriptions of aggressive behavior). This highlights the concern that the removal of PHI in free text may not be sufficient to make it anonymous. A reader who might have been present during a described event could potentially re-identify a person without direct identifiers, if the event was unique enough.

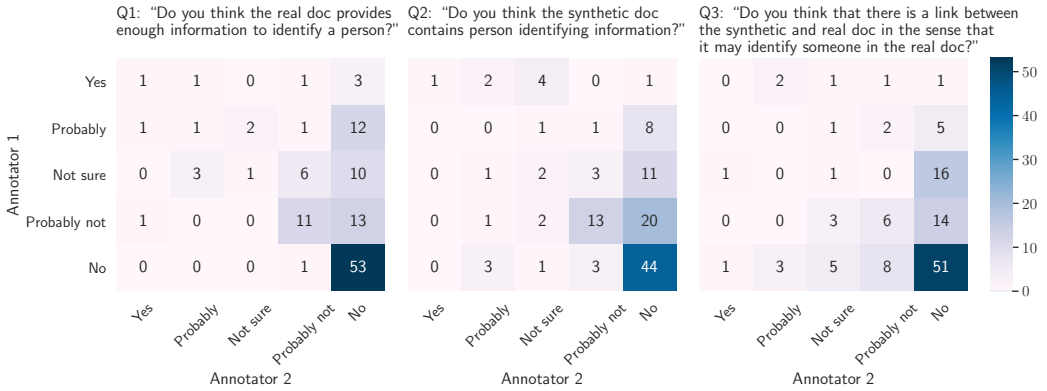

Figure 7. Inter-participant agreement (count of answer given) for the user study on privacy.

Question 2 (Information to Re-Identify a Person in Synthetic Document)

Similarly to the inter-participator agreement for question 1, Cohen's Kappa showed a fair agreement ($\kappa = 0.215$). Spearman's rank-order coefficient was $\rho = 0.4757$ ($p = 3.07 \times 10^{-8}$). The confusion matrix of participant responses in Figure 7 reveals that also for the synthetic documents shown, the contained information was often not considered person identifying. Some comments given for question 3 indicate that part of the reason may be the general incoherence of details that shows that the text is clearly fake and not about one specific person, thereby obfuscating which information is real and which PHI is related to it. For example, a text may reference several different names that do not fit together in context. This creates a privacy-protecting effect where information cannot be linked to one specific person. Furthermore, synthetic reports were often generic descriptions of days and medications without any identifiers. In cases where participants disagreed, but at least one answered with "Probably" or "Yes," reports were generally detailed and could contain person identifiers.

Question 3 (Identifying a Link between Real and Synthetic Document)

There was a slight agreement between participants ($\kappa = 0.063$ and $\rho = 0.4104$ with $p = 3 \times 10^{-6}$). In 42% of cases (51 of 122, Figure 7) both participants agreed that there was no link between the real and synthetic document. In cases where both participants agreed on the direction, but not strength of judgment and answered "Yes" or "Probably," the additional explanations revealed three categories of how synthetic text may identify someone from the real document:

1. **Contextual information was copied.** For example, the synthetic and real document described similar treatment, schedule or complications, sometimes with largely identical text including medical test results. One participant pointed out that the severity of this case would depend on the uniqueness of the medical test.
2. **Identifiers were copied.** For example, the same name(s) appeared in both documents. Unless contextual information was replicated, participants often disagreed on the severity of a potential privacy leak.
3. **The synthetic document acted as continuation of the real document with linked information**. Counterarguments to the existence of a privacy breach included inconsistencies in synthetic text that made it appear clearly fake (see Question 2) and generic content that made it hard to say whether a description was about the same person or not.

There were two examples in which participants agreed on a privacy breach. These contained specific descriptions of a diagnosis or situation that seemed unique enough to lead back to a person (e.g., someone dying soon, if in a non-dying population) and were copied from the original to a large extent. Interestingly, while the incoherence of certain synthetic text often added as protective factor for privacy, the effect may be reversed when a part of text is clearly fake and another part is clearly real, making it possible for a potential attacker to easily pick out copied information.

The findings of the privacy evaluation can be summarized as follows:

- In free text, the removal of PHI may not be sufficient to protect privacy when specific and rare events are described in detail.
- The mediocre quality of synthetic text often acted as protective factor by obfuscating what is real and what is fake.
- The largest cause of concern for privacy in this synthetic corpus is the existence of larger chunks of text that were copied from the real data, especially when rare events were described.

5. Implications and Outlook

In this section, we discuss the broader implications of our results and suggest avenues for future work to improve both utility and privacy of synthetic data.

5.1. Synthetic Data Generation and Text Quality

Controlling the distribution of annotations: We showed that it is possible to generate well-structured in-text annotations. However, we also observed that the distribution of tags depends on the chosen decoding method. This, in turn, had substantial impact on performance in downstream tasks. A desirable feature for generation methods is therefore the ability to control this distribution. Preliminary work in this direction, namely conditional transformer models [32,33], could be adapted for this purpose.

Increasing text diversity: Our experiments also revealed that text diversity has a significant impact on downstream task performance. In particular, we found that sampling methods provided both higher diversity and utility compared to beam search, which is in line with other results on open-ended text generation [24]. We think that future studies should strive to further increase the diversity of text. One promising direction is the so-

called "unlikelihood training" proposed by Welleck et al. [26], which increases diversity by changing the language modeling objective.

Improving text quality: The primary focus of this study was to generate documents with high utility for NLP models. Consequently, medical correctness and coherency was not formally evaluated. However, we found the coherence of synthetic documents to be mediocre. Related studies on generation of English EHR (mostly based on discharge letters in MIMIC-III) did not report such issues [7,8,13,14]. A key difference between MIMIC-III discharge letters and our Dutch healthcare corpus is the lack of clear structure and conformity in the Dutch corpus. To make methods for synthetic EHR generation applicable across healthcare, it would be beneficial to explore different pre-processing or model training strategies. One viable option could be to train separate models on subsets of notes that share structural properties.

Quantify how heuristic annotations influence downstream NER methods: We used a pre-trained method to automatically add in-text annotations to the language modeling data. While the pre-trained method showed high effectiveness ($F_1 = 0.895$, cf. Table 6) on highly similar data, we acknowledge that the annotations are imperfect. Therefore, it would be interesting to quantify how the accuracy of the in-text annotations influences the effectiveness of downstream NER models. As we are constrained by annotation resources, we leave the exploration of this idea to future research.

Transfer of method to other languages and domains: Instead of generating synthetic healthcare data for the Dutch language, the methodology of this research can also be used for different languages and text types: We trained the LSTM from scratch and since the architecture is not language specific, it may be applied to any sequence of tokens. Tokenization is language dependent, so pre-processing should be adjusted accordingly. We also fine-tuned the English pre-trained GPT-2 model and its tokenizer to learn Dutch, domain specific language and special annotations. This was possible, because there are similarities between Dutch and English. Sufficient similarity also exists with other languages, some of which GPT-2 has been adapted to previously (e.g., Italian [23,28]) and some open-source GPT-2 models pre-trained in different languages are openly available (e.g., a German pre-trained GPT-2 model: https://github.com/stefan-it/german-gpt2, accessed on 19 May 2021). GPT-2 is a "general purpose" model [6], because it can be adapted to different domains and language generation tasks, so cross-domain training is generally possible. While transfer of both LSTM and GPT-2 to other languages and domains is possible, applications that require generation of longer texts may require adjustments to the methodology (e.g., story generation [18]).

Support of other NLP downstream tasks: We investigated synthetic data generation in the context of de-identification. As de-identification is phrased as a standard NER task, we expect that our method generalizes well to other NER tasks. Future work is needed to investigate if language models can be adapted to produce other types of document metadata to support additional NLP downstream tasks such as classification.

5.2. Privacy of Synthetic Text

Privacy/utility trade-off: Our experiments showed that synthetic text does not need to be realistic for utility in downstream NER tasks. This could be exploited to improve the privacy protection. For example, a clearly incoherent combination of names within a document would obfuscate how pieces of information were originally linked. Therefore, future work could investigate how realistic synthetic text needs to be for a given downstream task. Prior work studied the trade-off between perplexity and privacy [7], where perplexity is a proxy for utility. This approach could be extended to take utility of synthetic text into account.

Expanding de-identification: Current approaches to text anonymization mostly define PHI as the 18-categories set out by the HIPAA regulation [34]. For example, documents in MIMIC-III are shared under the promise that all PHI have been removed and therefore protect privacy sufficiently. However, disregarding whether text was real or synthetic,

our user study identified certain aspects of notes which are not covered by automatic PHI extraction methods. Therefore, the common approach to protect privacy in natural language text might have to be re-evaluated and expanded to take, for example, specific descriptions of unusual events into account.

Embedding privacy: Given the examples of privacy leaks identified in the user study, it seemed that most would have been prevented if the model could not reproduce larger text chunks from a training EHR note. A way to ensure this from a mathematical perspective is to train the generative models with a differential privacy (DP) objective. The premise of DP is that no output could be directly attributed to a single training instance [2,7,19,35]. In this study, we consciously chose not to include DP to maximize the utility of the synthetic corpora for the downstream task, but we recommend that future research uses DP in order to minimize privacy risks.

Limitations of user study: While our user study provides insights into the privacy of synthetic records, it does not allow us to draw conclusions on the privacy of a synthetic corpus at large. To be able to publish synthetic corpora under the premise that they protect privacy of data subjects, principled ways of measuring the involved privacy risks are needed. Developing these approaches is an important direction for future work.

6. Conclusions

This paper proposes the use of language models to generate synthetic EHRs. By explicitly adding in-text annotations to the training data, the language models learn to produce artificial text that is automatically annotated for downstream NER tasks. Our experiments show that the synthetic data are of sufficient utility for downstream use in de-identification. In particular, a de-identification method trained on synthetic data outperforms a rule-based method. Moreover, augmenting real data with synthetic data further improves the recall of the method at no additional costs or manual annotation effort. We find that the LSTM-based method produces synthetic text with higher utility in the downstream task compared to GPT-2. This is despite the fact that GPT-2 texts are more coherent. This suggests that coherence is not required for synthetic text to be useful in downstream NER tasks. We furthermore evaluate privacy of the generated synthetic data using text-proximity metrics and conduct a user study. We find that the synthetic documents are not free of privacy concerns because language models replicated potentially identifying chunks of real EHRs. This shows that additional work is needed before synthetic EHRs can be used as an anonymous alternative to real text in data sharing settings.

Author Contributions: Conceptualization, C.A.L., J.T., D.T., C.S.; methodology C.A.L., J.T., D.T., C.S.; software, C.A.L.; writing—original draft, C.A.L., J.T.; writing—review and editing, J.T., D.T., C.S.; supervision, C.S. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge support by the Open Access Publication Fund of the University of Duisburg-Essen.

Data Availability Statement: The data used in this study was pseudonymized for privacy protection. We received approval for the collection and use of the dataset from the privacy board of Nedap Healthcare. Because of privacy regulations, the dataset cannot be made publicly available.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Fine-Tuning English GPT-2 to Dutch Language

This appendix provides additional information on how we adapted the English GPT-2 model to Dutch healthcare data. At the time when we conducted this research, no study reported the code or a detailed strategy to adapt GPT-2 for a non-English purpose. Therefore, we followed the approach described by Pierre Guilliou adapting GPT-2 to Portuguese. The report can be found here: https://medium.com/@pierre_guillou/faster-than-training-from-scratch-fine-tuning-the-english-gpt-2-in-any-language-with-hugging-f2ec05c98787, accessed on 19 May 2021. The approach is similar to the work (published later) by de Vries and Nissim [28]. Below, we outline how the tokenizer was extended to the Dutch vocabulary and provide the fine-tuning steps in Table A1.

1. Settings of the Byte-Pair Encoding (BPE) tokenizer: Initial size equals to vocabulary length $|V|$ of English pre-trained GPT-2 tokenizer. Minimum token frequency is set to 2. We add a prefix space as well as special tokens for PHI tags and paragraph delimiters (e.g., `<PAR>`, `<NameSTART>`, `<NameEND>`). Sequences are truncated with a maximum sequence length of 1024. Padding token is set to `<|endoftext|>`.
2. New word-token-embedding matrix is initialized by copying English embeddings for overlapping terms. New (Dutch) terms are subsequently added to the embedding matrix and initialized with the mean of the English embedding matrix.
3. Model is fine-tuned according to the steps in Table A1.

Table A1. Fine-tuning steps of GPT-2. The fastai library was used to split layer groups and to fine-tune the model with one-cycle policy [29]. Differential learning for several layers is applied by passing an array of learning rates `fit_one_cycle()` (https://docs.fast.ai/callback.schedule.html#Learner.fit_one_cycle, accessed on 19 May 2021). Training parameters from Pierre Guillou (https://medium.com/@pierre_guillou/faster-than-training-from-scratch-fine-tuning-the-english-gpt-2-in-any-language-with-hugging-f2ec05c98787, accessed on 19 May 2021).

Step	Layer Groups	Learning Rates
1.	All frozen, fitted for 1 cycle	`fit_one_cycle(1, 2e-3)`
2.	Last two layer groups unfrozen. Fitted for 1 cycle: Decoder blocks 8–11, Vocabulary embedding, Positioning embedding, LayerNorm at model output	`fit_one_cycle(1, slice(1e-3/(2.6**4),1e-3))`
3.	Last three layer groups unfrozen. Fitted for 1 cycle: Previous layers, Decoder blocks 4–7	`fit_one_cycle(1, slice(5e-4/(2.6**4),5e-4))`
4.	All layer groups unfrozen. Fitted for 2 cycles: Previous layers, Decoder blocks 0–3	`fit_one_cycle(2, slice(1e-4/(2.6**4),1e-4))`

Appendix B. Distribution of PHI Tags in Synthetic Corpora

We provide the absolute number of PHI tags per corpus in Table A2 and compare the distribution of tags across corpora in Figure A1. Furthermore, Figure A2 quantifies how much the PHI distribution in each corpus differs from the PHI distribution of the language modeling data (raw numbers for Figure 5).

Table A2. Absolute PHI counts in all corpora. The "LM Corpus" is used to develop the language models. "LM Corpus" counts are reproduced from Table 1 and "NUT" counts from [10].

PHI Tag	LM Corpus	LSTM-p	LSTM-Temp	GPT-p	GPT-Beam	NUT
Name	782,499	20,697	19,839	34,764	6797	9558
Date	202,929	4270	4240	19,879	12,825	3676
Initials	181,811	4038	4166	11,337	2771	778
Address	46,387	1244	1220	6834	299	748
Care Inst.	38,669	1006	985	8537	437	997
Org.	37,284	1091	1041	11,885	1100	712
Location	6977	115	117	1486	56	242
Phone/Fax	3843	45	27	4539	74	97
Age	3350	40	60	416	12	175
Email	2539	40	26	4298	55	95
Hospital	2425	44	46	191	34	92
Profession	537	4	5	32	0	122
URL/IP	326	4	2	723	9	23
ID	232	0	1	200	1	114
Other	105	1	1	0	0	33
SSN	6	0	0	0	0	2
Total	1,309,919	32,639	31,776	105,121	24,470	17,464

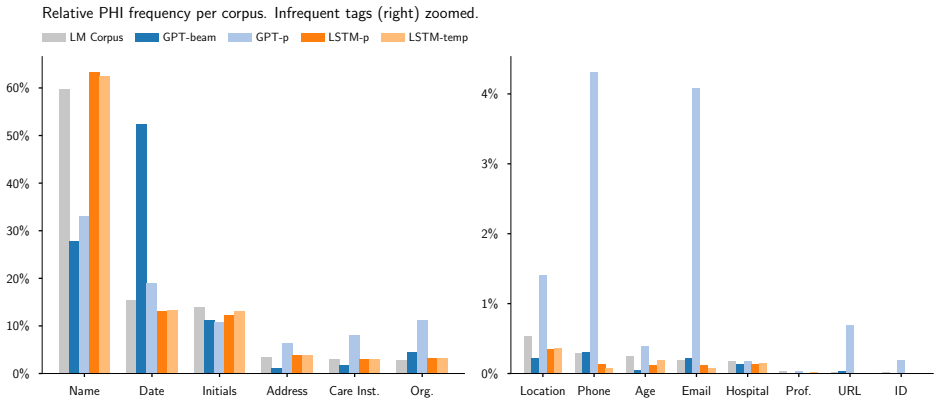

Figure A1. PHI distribution of the synthetic corpora compared to the language modeling corpus.

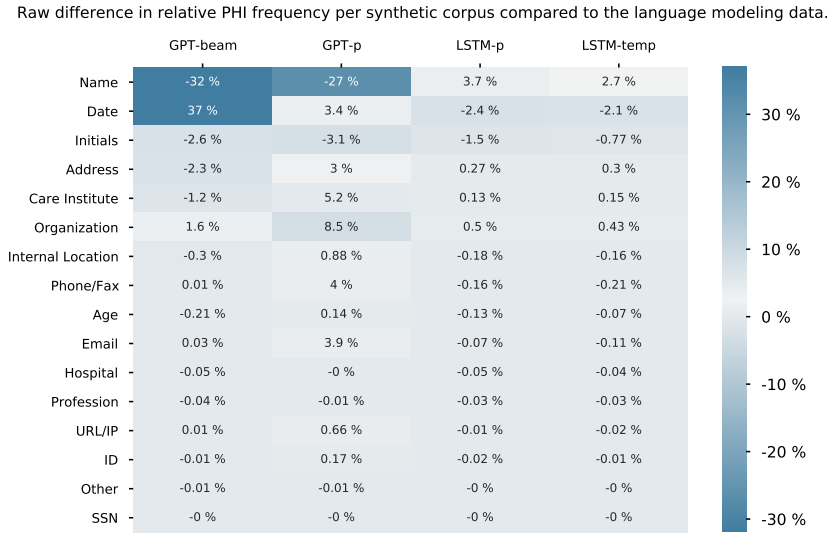

Figure A2. Raw increase/decrease in relative PHI frequency per synthetic corpus compared with the PHI distribution of the language modeling corpus. Raw numbers for Figure 5.

Appendix C. Additional Examples of Synthetic Documents

Additional examples for each text generation method are shown in Table A3.

Table A3. Example documents generated by each method with manual English translation. For all examples, we used the same input prompt "Mevr. was vanavond", which translates to "Mrs. was [...] tonight". We transformed <PAR> back into line breaks, but otherwise left in-text annotations as they were generated.

Model	Example (Dutch Original)	Example (English Translation)
LSTM-p	Mevr. was vanavond veel aan het slaan in de huiskamer. Mevr. vertelde dat ze het al even kan laten merken. Op dat moment moest zij even naar het toilet omdat ze kon plassen. Mevr. voelde zich in het inco broekje niet een lange broek aan, ik heb met collega een mail naar de voedingsadministratie gedaan.	Mrs. was beating much in the living room tonight. Mrs. told that she could let notice it already. At that moment she had to go the toilet because she could pee. Mrs. felt herself in the inco pants not long pants wearing, I have send an email to the food administration with a colleague.
LSTM-temp	Mevr. was vanavond niet op de Huismus. Moest het haar nog een keertje uit zetten. Ik heb mevr. haar appartement laten zien en met 2 poten op de stoel gelopen. Mevr. heeft niet veel geslapen. Mevr. heeft vannacht niet verschoond en deze ging nu altijd bij, zat prima in haar stoel.	Mrs. was not at the Huismus tonight. Had to turn her off for once. I showed mrs. her apartment and walked with 2 legs at the chair. Mrs. has not slept much. Mrs. did not change tonight and this now always went with, sat in her chair fine.
GPT-p	Mevr. was vanavond rustig aanwezig. Mw. heeft vanavond goed gegeten en gedronken. <NameSTART> Lucy <NameEND> Rapp. <NameSTART> B. Greenwood <NameEND> broecks Dag <NameSTART> Barbara <NameEND>, Ik heb net contact gehad met <NameSTART> Alex <NameEND>. <Organization_CompanySTART> de Zonnebloem <Organization_CompanyEND> <NameSTART> Jane <NameEND> is op de hoogte van de situatie.	Mrs. was quietly present tonight. Mrs. has eaten and drank well tonight. <NameSTART> Lucy <NameEND> Rep. <NameSTART> B. Greenwood <NameEND> broecks Hello <NameSTART> Barbara <NameEND>, I have just had contact with <NameSTART> Alex <NameEND>. <Organization_CompanySTART> de Zonnebloem <Organization_CompanyEND> <NameSTART> Jane <NameEND> is aware of the situation.
GPT-beam	Mevr. was vanavond rustig aanwezig. Mevr. heeft goed gegeten en gedronken. Mevr. is om 21.00 uur naar bed geholpen. mevr. gaf aan erg moe te zijn en graag naar bed te willen. Mevr. is om 22.30 uur in bed geholpen en ligt tot nu toe nog te slapen. <DateSTART> Zondag <DateEND> komt mevr. weer naar de dagbesteding. <unk> Mevr. geeft aan het erg naar haar zin te hebben gehad.	Mrs. was quietly present tonight. Mrs. has eaten and drank well. Mrs. was helped to bed at 9 pm. Mrs. indicated to be very tired and would like to go to bed. Mrs. was helped to bed at 10.30 pm and is still sleeping until now. <DateSTART> Sunday <DateEND> mrs. will come to the daytime activities. Mrs. indicated that she had a great time.

Appendix D. Privacy User Study: Annotation Guidelines and Data Sampling

We provide annotation guidelines in Figure A4. Below, we outline the steps to filter a sample of real-synthetic document pairs SR for presentation to participants. We denote a synthetic document as $s \in S$ and a real document as $r \in R$.

1. Remove duplicates: for the same document s, ROUGE-3 and ROUGE-5 may retrieve the same document r.
2. Sort the synthetic documents by ROUGE-3 and ROUGE-5 recall and keep the top-100 of both lists. (The top 100 ROUGE-3 recall scores were between 0.18 and 1.0 with an average of 0.307 and a median of 0.233. The top 100 ROUGE-5 recall scores were between 0.111 and 1.0 with an average of 0.236 and a median of 0.164.) The idea is that we investigate high risk documents with highly similar counterparts among the real data. Add these documents to SR.
3. For the remaining documents in SR, retrieve the most similar document with BM25.
4. Remove documents longer than 1000 characters to control annotation effort.

5. Remove documents that had a high overlap due to structural elements (e.g., <PAR> token or punctuation).

Appendix E. Evaluating the Impact of the Synthetic Dataset Size

The effectiveness of a downstream machine learning method necessarily depends on the number of (synthetic) training examples. For simplicity, we fixed the size of the synthetic datasets across all our experiments (cf. Section 3.3.3). To analyze if it would be beneficial to increase/decrease the size of the synthetic corpora, we trained de-identification models on subsets of the data. Figure A3 shows the entity-level F1-score for varying training set sizes. We find that the learning curve flattens at around 70% of the training data, indicating that there is little benefit to generate even larger synthetic corpora. Due to computational constraints, we limited this experiment to one synthetic corpus (LSTM-p).

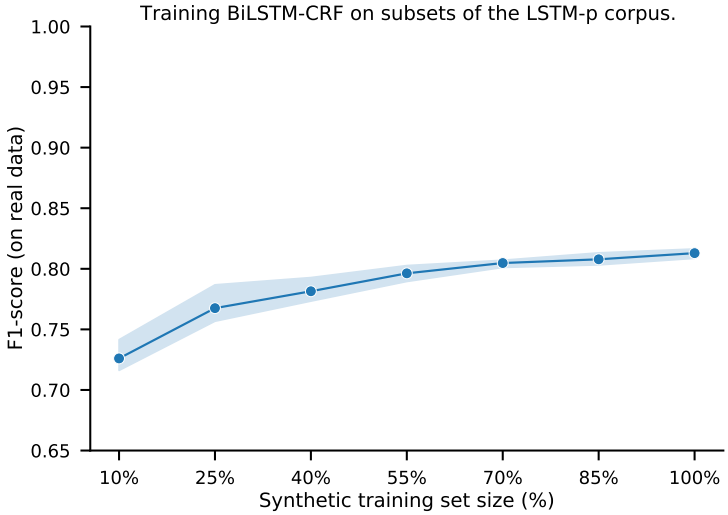

Figure A3. Entity-level F1-score for varying LSTM-p training set sizes. The full training set (100%) consists of all training and validation documents in LSTM-p. The F1-score is measured on the *NUT* test set. For each subset size, we train/test each model 3 times. The line shows the averaged scores along with the 95% confidence interval.

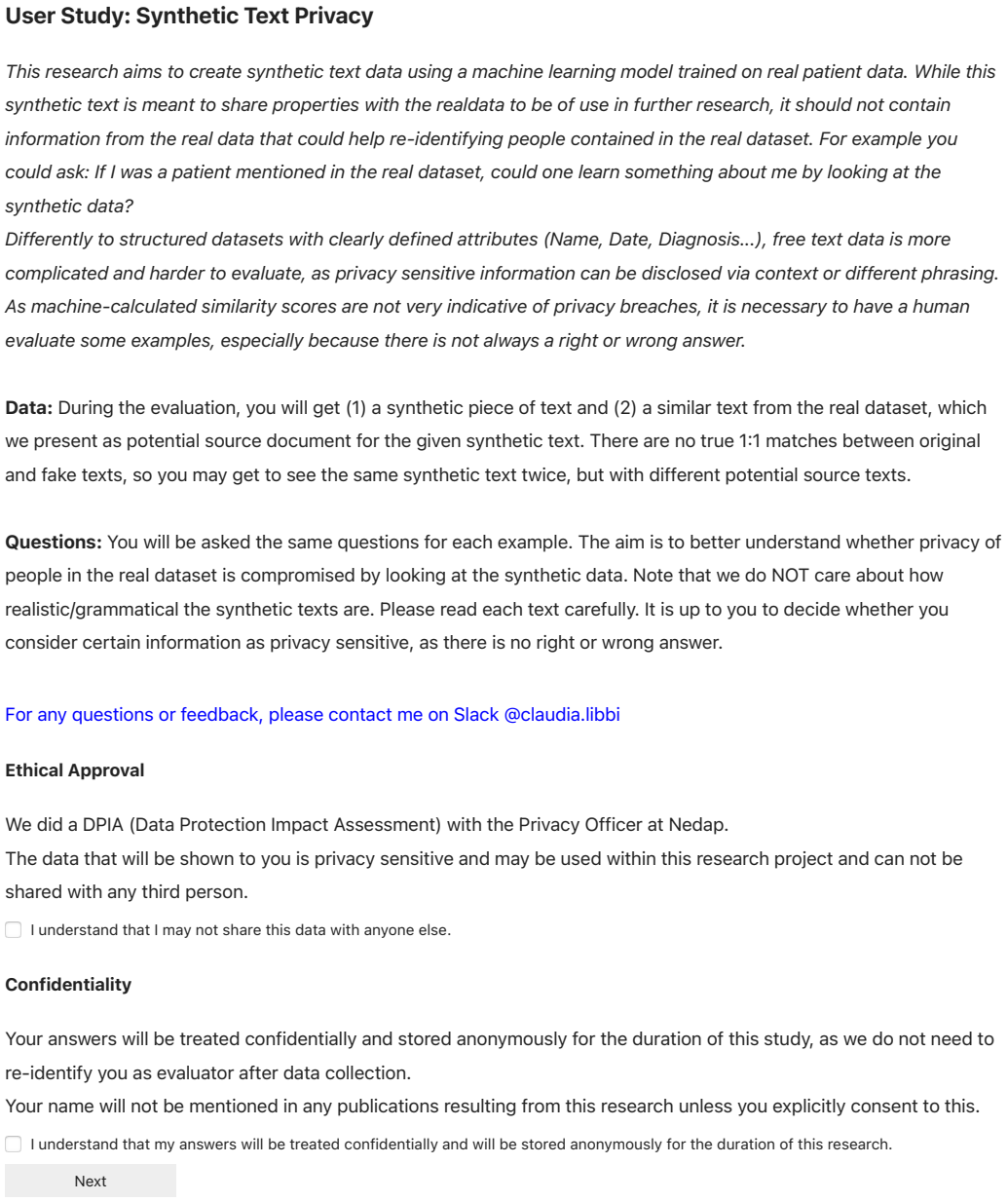

Figure A4. Annotation guidelines for the privacy user study.

References

1. Wang, Y.; Wang, L.; Rastegar-Mojarad, M.; Moon, S.; Shen, F.; Afzal, N.; Liu, S.; Zeng, Y.; Mehrabi, S.; Sohn, S.; et al. Clinical information extraction applications: A literature review. *J. Biomed. Inform.* **2018**, *77*, 34–49. [CrossRef] [PubMed]
2. Bellovin, S.M.; Dutta, P.K.; Reitinger, N. Privacy and synthetic datasets. *Stan. Tech. L Rev.* **2019**, *22*, 1. [CrossRef]
3. Rankin, D.; Black, M.; Bond, R.; Wallace, J.; Mulvenna, M.; Epelde, G. Reliability of supervised machine learning using synthetic data in health care: Model to preserve privacy for data sharing. *JMIR Med. Inform.* **2020**, *8*, e18910. [CrossRef] [PubMed]
4. Wang, L.; Liu, J.; Liu, J. Investigating Label Bias in Beam Search for Open-ended Text Generation. *arXiv* **2020**, arXiv:2005.11009.
5. El Emam, K.; Mosquera, L.; Bass, J. Evaluating Identity Disclosure Risk in Fully Synthetic Health Data: Model Development and Validation. *J. Med. Internet Res.* **2020**, *22*, e23139. [CrossRef] [PubMed]
6. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
7. Melamud, O.; Shivade, C. Towards Automatic Generation of Shareable Synthetic Clinical Notes Using Neural Language Models. In Proceedings of the 2nd Clinical Natural Language Processing Workshop, Minneapolis, MN, USA, 7 June 2019; pp. 35–45.
8. Amin-Nejad, A.; Ive, J.; Velupillai, S. Exploring Transformer Text Generation for Medical Dataset Augmentation. In Proceedings of the 12th Language Resources and Evaluation Conference, LREC Marseille, France, 11–16 May 2020; pp. 4699–4708.

9. Meystre, S.M. De-identification of Unstructured Clinical Data for Patient Privacy Protection. In *Medical Data Privacy Handbook*; Gkoulalas-Divanis, A., Loukides, G., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2015; pp. 697–716.
10. Trienes, J.; Trieschnigg, D.; Seifert, C.; Hiemstra, D. Comparing Rule-based, Feature-based and Deep Neural Methods for De-identification of Dutch Medical Records. In Proceedings of the ACM WSDM 2020 Health Search and Data Mining Workshop, co-located with the 13th ACM International WSDM Conference, HSDM@WSDM 2020, Houston, TX, USA, 6–9 February 2020; Volume 2551, pp. 3–11.
11. Lin, C.Y. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*; Association for Computational Linguistics: Barcelona, Spain, 2004; pp. 74–81.
12. Manning, C.D.; Raghavan, P.; Schütze, H. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008.
13. Liu, P.J. Learning to Write Notes in Electronic Health Records. *arXiv* **2018**, arXiv:1808.02622.
14. Wang, Z.; Ive, J.; Velupillai, S.; Specia, L. Is artificial data useful for biomedical Natural Language Processing algorithms? In Proceedings of the 18th BioNLP Workshop and Shared Task, BioNLP@ACL 2019, Florence, Italy, 1 August 2019; pp. 240–249.
15. Ive, J.; Viani, N.; Kam, J.; Yin, L.; Verma, S.; Puntis, S.; Cardinal, R.N.; Roberts, A.; Stewart, R.; Velupillai, S. Generation and evaluation of artificial mental health records for Natural Language Processing. *NPJ Digit. Med.* **2020**, *3*, 1–9. [CrossRef] [PubMed]
16. de Vries, W.; van Cranenburgh, A.; Bisazza, A.; Caselli, T.; van Noord, G.; Nissim, M. BERTje: A Dutch BERT Model. *arXiv* **2019**, arXiv:1912.09582.
17. Delobelle, P.; Winters, T.; Berendt, B. RobBERT: A Dutch RoBERTa-based Language Model. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16–20 November 2020; pp. 3255–3265.
18. Peng, N.; Ghazvininejad, M.; May, J.; Knight, K. Towards Controllable Story Generation. In Proceedings of the First Workshop on Storytelling, Grenoble, France, 26 March 2018; pp. 43–49.
19. Yoon, J. End-to-End Machine Learning Frameworks for Medicine: Data Imputation, Model Interpretation and Synthetic Data Generation. Ph.D. Thesis, UCLA, Shenzhen, China, 2020.
20. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
21. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
22. Zoph, B.; Yuret, D.; May, J.; Knight, K. Transfer Learning for Low-Resource Neural Machine Translation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, TX, USA, 1–5 November 2016; pp. 1568–1575.
23. Mattei, L.D.; Cafagna, M.; Dell'Orletta, F.; Nissim, M.; Guerini, M. GePpeTto Carves Italian into a Language Model. In Proceedings of the Seventh Italian Conference on Computational Linguistics, CLiC-it 2020, Bologna, Italy, 1–3 March 2020; Volume 2769.
24. Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; Choi, Y. The Curious Case of Neural Text Degeneration. In Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020.
25. Nadeem, M.; He, T.; Cho, K.; Glass, J. A Systematic Characterization of Sampling Algorithms for Open-ended Language Generation. In Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, Online, 17–18 October 2020; pp. 334–346.
26. Welleck, S.; Kulikov, I.; Roller, S.; Dinan, E.; Cho, K.; Weston, J. Neural Text Generation With Unlikelihood Training. In Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020.
27. Stubbs, A.; Uzuner, Ö.; Kotfila, C.; Goldstein, I.; Szolovits, P. Challenges in Synthesizing Surrogate PHI in Narrative EMRs. In *Medical Data Privacy Handbook*; Gkoulalas-Divanis, A., Loukides, G., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2015; pp. 717–735.
28. de Vries, W.; Nissim, M. As Good as New. How to Successfully Recycle English GPT-2 to Make Models for Other Languages. *arXiv* **2020**, arXiv:2012.05628.
29. Smith, L.N.; Topin, N. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates. *arXiv* **2018**, arXiv:1708.07120.
30. Menger, V.; Scheepers, F.; van Wijk, L.M.; Spruit, M. DEDUCE: A pattern matching method for automatic de-identification of Dutch medical text. *Telemat. Inform.* **2018**, *35*, 727–736. [CrossRef]
31. Choi, E.; Biswal, S.; Malin, B.A.; Duke, J.; Stewart, W.F.; Sun, J. Generating Multi-label Discrete Patient Records using Generative Adversarial Networks. In Proceedings of the Machine Learning for Health Care Conference, MLHC 2017, Boston, MA, USA, 18–19 August 2017; Volume 68, pp. 286–305.
32. Keskar, N.S.; McCann, B.; Varshney, L.R.; Xiong, C.; Socher, R. CTRL: A Conditional Transformer Language Model for Controllable Generation. *arXiv* **2019**, arXiv:1909.05858.
33. Dathathri, S.; Madotto, A.; Lan, J.; Hung, J.; Frank, E.; Molino, P.; Yosinski, J.; Liu, R. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. In Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020.

34. HIPAA. Guidance Regarding Methods for De-Identification of Protected Health Information in Accordance with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule. Available online: https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html (accessed on 19 May 2021).
35. Hittmeir, M.; Ekelhart, A.; Mayer, R. Utility and Privacy Assessments of Synthetic Data for Regression Tasks. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 5763–5772.