How to keep text private? A systematic review of deep learning methods for privacy-preserving natural language processing

A Preprint

Samuel Sousa
Know-Center GmbH &
Institute of Interactive Systems and Data Science
Graz University of Technology
Graz, Austria
ssousa@know-center.at

Roman Kern
Know-Center GmbH &
Institute of Interactive Systems and Data Science
Graz University of Technology
Graz, Austria
rkern@know-center.at

Abstract

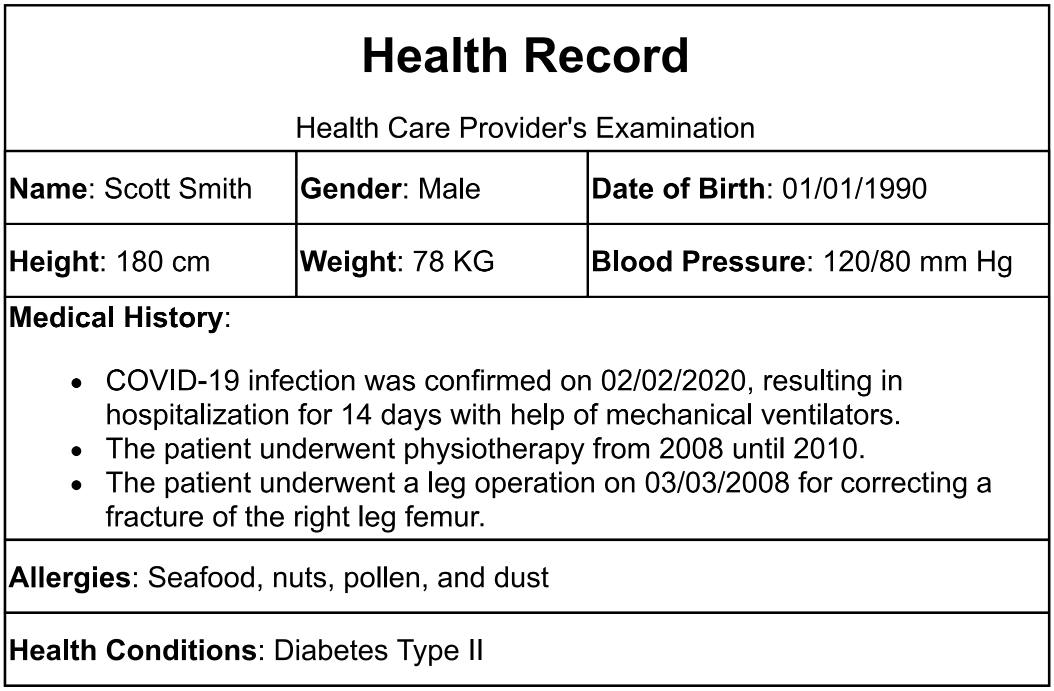
Deep learning (DL) models for natural language processing (NLP) tasks often handle private data, demanding protection against breaches and disclosures. Data protection laws, such as the European Union's General Data Protection Regulation (GDPR), thereby enforce the need for privacy. Although many privacy-preserving NLP methods have been proposed in recent years, no categories to organize them have been introduced yet, making it hard to follow the progress of the literature. To close this gap, this article systematically reviews over sixty DL methods for privacy-preserving NLP published between 2016 and 2020, covering theoretical foundations, privacy-enhancing technologies, and analysis of their suitability for real-world scenarios. First, we introduce a novel taxonomy for classifying the existing methods into three categories: data safeguarding methods, trusted methods, and verification methods. Second, we present an extensive summary of privacy threats, datasets for applications, and metrics for privacy evaluation. Third, throughout the review, we describe privacy issues in the NLP pipeline in a holistic view. Further, we discuss open challenges in privacy-preserving NLP regarding data traceability, computation overhead, dataset size, the prevalence of human biases in embeddings, and the privacy-utility tradeoff. Finally, this review presents future research directions to guide successive research and development of privacy-preserving NLP models.

Keywords Deep learning, Privacy, Natural language processing, Differential privacy, Homomorphic encryption, Searchable encryption, Federated learning

1 Introduction

Privacy is the ability to control the extent of personal matters an individual wants to reveal (Westin 1968). It is protected by regulations in many countries around the planet, like the European Union (EU)'s General Data Protection Regulation (GDPR) (Commission 2018), in order to protect the security, integrity, and freedom of people. For instance, the GDPR establishes guidelines for the collection, transfer, storage, management, processing, and deletion of personal data within the EU. Penalties and fines are also applicable in case of misbehavior or non-compliance with legal terms. The approval of data protection laws has driven an increased need for privacy-enhancing technologies (PETs), which control the amount of existing personal data, limiting it or ridding it, as well as its processing, without losing system functionalities (Van Blarkom et al. 2003). Therefore, PETs constitute the cornerstone for the privacy preservation of personal data.

Data from several domains, such as finance (Han et al. 2019), documents (Eder et al. 2019), bio-medicine (Dernoncourt et al. 2017), social media (Blodgett and O'Connor 2017; Salminen et al. 2020), and images (He et al. 2019; Sánchez et al. 2018), inherently present sensitive content which must be protected. Such sensitive attributes include a person's



Health Record

Health Care Provider's Examination

Name: Scott Smith	**Gender**: Male	**Date of Birth**: 01/01/1990
Height: 180 cm	**Weight**: 78 KG	**Blood Pressure**: 120/80 mm Hg

Medical History:

- COVID-19 infection was confirmed on 02/02/2020, resulting in hospitalization for 14 days with help of mechanical ventilators.
- The patient underwent physiotherapy from 2008 until 2010.
- The patient underwent a leg operation on 03/03/2008 for correcting a fracture of the right leg femur.

Allergies: Seafood, nuts, pollen, and dust

Health Conditions: Diabetes Type II

Figure 1 Pieces of private textual information in a health record

identity, age, gender, home address, location, income, marital status, ethnicity, political and societal views, health conditions, pictures, as well as any other traits that allow their identification or have the potential to harm their safety or reputation (Alawad et al. 2020). In text data, private information can be found in many features derived from the text content in documents, e-mails, chats, online comments, medical records, and social media platforms. Figure 1 depicts some pieces of private information in text data as the fields of a health record. These fields contain demographic attributes (e.g., gender and age), physical characteristics (e.g., height, weight, and blood pressure), history of medical treatments, allergies, a person's full name, and health conditions. A model for predicting diseases from such a health record must not reveal the identity and health information of the hospital patient who generated it.

The preservation of privacy is a bottom line for developing new deep learning (DL) methods in scenarios that feature sensitive or personal data alongside threats of breaches. In addition, deep neural network models are often a target for attacks that aim at recovering data instances used for training, especially reverse engineering (Li et al. 2018), membership inference (Pan et al. 2020), and property inference (Ganju et al. 2018). PETs (Van Blarkom et al. 2003), which are computational techniques to manage privacy risks, emerged thereon, covering all phases of the model pipeline, from data pre-processing up to the validation of results. In the past few years, PETs have drawn attention in the literature since DL architectures have often been applied to private data combined with them (Boulemtafes et al. 2020). Well-known PETs for DL include fully homomorphic encryption (FHE) (Gentry 2009), differential privacy (DP) (Dwork 2008), adversarial learning (Goodfellow et al. 2014), federated learning (FL) (Yin et al. 2020), multi-party computation (MPC) (Goldreich 1998), and secure components (Jia et al. 2013). Although these technologies protect privacy from disclosures and attacks, they come with utility-efficiency tradeoffs. For instance, adding noise to a model's training data often degrades its performance. Running DL models on encrypted data can also be memory-demanding and time-consuming.

Recent advances in the field of natural language processing (NLP) have been provided by DL methods that steadily led to outstanding results for tasks, such as automatic question-answering (Minaee and Liu 2017), dependency parsing (Kiperwasser and Goldberg 2016), machine translation (Vaswani et al. 2018), natural language understanding (Sarikaya et al. 2014), text classification (Liu et al. 2017a), and word sense disambiguation (Sousa et al. 2020). However, DL models trained or used for inference on sensitive text data may be susceptible to privacy threats where the learning setting features more than a single computation party or the data is outsourced (Boulemtafes et al. 2020). Therefore, privacy is an essential requirement for developing NLP applications for personal and private data.

Preserving the privacy of text data is a tough challenge due to the hardness of identifying sensitive attributes in text alongside the computational costs PETs may demand. In some cases, removing private attributes does not hinder their inference from their surrounding context. For instance, location names can be placed near descriptions that allow their identification. As a response to a large number of privacy-related issues in NLP, a broad separate literature has developed as an active research topic. Despite its broadness, the literature on privacy-preserving NLP does not present a precise categorization of approaches regarding DL and PETs to provide an overview of the main topics and technologies to both the scientific community and industry practitioners. Table 1 lists all the abbreviations we make use of throughout this review.

Table 1 Abbreviations used throughout this review

Abbreviation	Description
AA	Authorship attribution
AI	Artificial intelligence
BiLSTM	Bidirectional long short-term memory
CNN	Convolutional neural network
DA	Domain adaptation
DL	Deep learning
DP	Differential privacy
EU	European Union
FHE	Fully-homomorphic encryption
FL	Federated learning
GDPR	General Data Protection Regulation
HIPAA	Health Insurance Portability and Accountability Act
LSTM	Long short-term memory
MPC	Multi-party computation
NLP	Natural language processing
PETs	Privacy-enhancing technologies
PHI	Protected health information
RNN	Recurrent neural network
SE	Searchable encryption
SEAT	Sentence Encoder Association Test
TL	Transfer learning
WEAT	Word Embedding Association Test

1.1 Related surveys and reviews

Privacy preservation has multiple perspectives involving privacy metrics, DL models, NLP tasks, threats, computation scenarios, and PETs. This work lies at the intersection of these perspectives, aiming to bridge the gap between them. For this reason, we review the literature on privacy, DL, and NLP from 2016 to 2020, to provide an extensive review of the privacy-preserving DL methods for NLP and draw attention to the privacy issues NLP data may face. Some recently published survey papers have brought overviews on privacy metrics, privacy-preserving DL, security attacks, security issues, biases in NLP, privacy on social media, and adversarial threats in text data, as Table 2 shows. Each work in the table introduces different perspectives on privacy preservation, yet apart from a broad unified review for NLP applications based on DL models. Moreover, the papers we review rarely overlap those reviewed by the works in Table 2; hence, highlighting the comprehensive coverage of our review.

A primary problem for privacy regards its measurement. Many metrics have been proposed, so Wagner and Eckhoff (2018) review more than 80 metrics for measuring privacy, introducing a novel classification determined by four determinants. First, the capabilities that an adversary model is likely to present, such as information about a probability distribution. Second, the data sources, especially public and private data. Third, the expected inputs, e.g., parameters. Finally, properties privacy metrics measure, including data similarity, error, information gain, or information loss. A procedure for choosing the most suitable privacy measures for different backdrops is also provided, followed by the advice on choosing more than a single metric to cover a larger number of privacy aspects. In the end, the authors point out the need for further research on privacy metrics aggregation concerning cases in which personal privacy is affected by other parties.

Humbert et al. (2019) survey risks and solutions for interdependent privacy, proposing categories for both. For example, when users of different e-mail providers engage in message exchange, the history of such communication is stored

Table 2 Overview of related surveys on privacy for DL and NLP

Work	Year	Contributions
Technical privacy metrics: A systematic survey (Wagner and Eckhoff 2018).	2018	Summary and classification of over 80 privacy metrics, theoretical foundations, application domains, examples in context, and a methodology to determine suitable metrics for application scenarios.
Mitigating gender bias in natural language processing: Literature review (Sun et al. 2019).	2019	Literature review of recent works on recognition and mitigation of algorithmic gender bias in the NLP field and discussion aiming to point out drawbacks and gaps for future research.
Text analysis in adversarial settings: Does deception leave a stylistic trace? (Gröndahl and Asokan 2019).	2019	Study of changes in authors' writing styles to detect deceptive text, analysis of whether authorship identification techniques encompass privacy threats, description of backdrops for attacks, the outline of style obfuscation and imitation methods, and discussion on the traceability of style obfuscation techniques.
A survey on interdependent privacy (Humbert et al. 2019).	2019	Insights on privacy and interdependent privacy, highlighting how different scientific communities approach these topics; review of tools and theoretical frameworks for research; and the categorization of risks, concerns, and solutions.
A review of privacy-preserving techniques for deep learning (Boulemtafes et al. 2020).	2020	A multi-level taxonomy of privacy-preserving techniques for DL, followed by their performance analysis.
The AI-based cyber threat landscape: A survey (Kaloudi and Li 2020).	2020	The map of cyber-attacks adopting AI techniques onto a framework for the classification of dishonest AI uses. This work also presents a basis for their detection in future events. An attack backdrop on a smart grid infrastructure is also included to illustrate how to apply the proposed framework.
Adversarial attacks on deep-learning models in natural language processing: A survey (Zhang et al. 2020).	2020	The outline, discussion, and taxonomy of works on generating adversarial examples for NLP applications based on DL models.

on the servers of both e-mail providers. In this case, the privacy of an e-mail user depends on the other's actions, like leaking the exchanged messages or keeping them secret. The risks are sorted in the survey according to the data type they arise from, such as demographic, genomic, multimedia, location, and aggregated data. Similarly, the solutions for interdependent privacy are split into two groups based on their complexity. On the one hand, simple manual user actions, like limiting data visibility for other users, compose the group of non-technical solutions (also referred to as social solutions). On the other hand, solutions that rely on computational tools, software architectures, encryption, and similar are grouped into technical solutions. The authors also argue that research approaches for interdependent privacy should lean mostly on principles rather than data dependency.

In DL, a total of 45 privacy-preserving solutions are reviewed by Boulemtafes et al. (2020) and then organized as a four-level taxonomy. The first level regards privacy-preserving tasks, namely model learning, analysis, and model releasing. Level 2 of the taxonomy refers to the learning paradigm of the reviewed techniques, like individual or collaborative learning. The third level draws up differences between server-based and server-assisted approaches. Last, the fourth taxonomy level classified privacy-preserving methods based on technological concepts, such as encryption or MPC. Kaloudi and Li (2020) also cover DL architectures in their survey of cyber-attacks based on artificial intelligence (AI) techniques, such as password crackers which use self-learning for brute-force attacks. However, the authors focus on malicious uses of such neural networks, presenting a classification of attacking threats as a future prediction framework. Consequently, this new framework can be helpful for implementing new safeguarding measures against the identified risk scenarios.

NLP data and tasks also encompass privacy-related issues requiring special algorithmic solutions. Sun et al. (2019) review studies on identifying and mitigating gender bias in text data tasks. The authors propose four categories of representation bias, relying on problems noticed in the datasets, such as derogatory terms, societal stereotypes, and under-representation of some groups, and models, like their inaccuracies for sensitive tasks. Bias arising from text data can easily be embedded into vector representations of words and consecutively propagated to downstream tasks so that bias removal methods are demanded. The conclusions of this review are four-fold. First, it is observed that debiasing methods are usually implemented as end-to-end solutions whose interactions between their parts are still unknown. Second, debiasing methods' generalization power is questioned since they are mostly tested on narrow NLP tasks. Further, the authors have argued that some of these methods may introduce noise into NLP models, leading to drops in performance. Last, it is noticed that hand-crafted debiasing techniques may be inadvertently biased by their developers.

An outstanding security issue in NLP regards deceptive language (Mihalcea and Strapparava 2009). Gröndahl and Asokan (2019) provide an overview of empirical works on the detection of deceptive text. The authors first target misleading content, i.e., detecting dishonest text based on the author's writing style. Second, they focus on adversarial stylometry, which is considered a type of deceptive text since truthful information pieces in the data were replaced with anonymized versions. It consists of a method against deanonymization attacks. Finally, they conclude that stylometry analysis is efficient in predicting deceptive text when the training and test domains are adequately related. However, deanonymization attacks will continue to be a rampant and growing privacy threat. As a result, manual style obfuscation approaches are expected to solve more efficiently than automatic ones.

Beyond deanonymization attacks, text data can also be used for adversarial attacks against DL models. In these attacks, data samples are generated with small perturbations but can dupe DL models and prompt false predictions. Zhang et al. (2020) overview 40 works that addressed adversarial attacks towards DL models for NLP tasks, such as text classification, machine translation, and text summarization. To split the works into categories in a taxonomy, the authors consider five viewpoints in both model and semantic perspectives. First, knowledge of the attacked model at the time of the attack. Second, NLP applications. Third, the target for the attack, e.g., incorrect predictions or specific results. Further, the granularity level of the attack which ranges from character level to sentence level. Finally, the attacked DL model, such as convolutional neural network (CNN), recurrent neural network (RNN), and autoencoders. State-of-the-art methods, general NLP tasks, defenses, and open challenges are also discussed. The authors then raise the issue that adversarial examples can be used for membership inference attacks, which reconstruct the original data samples used for training DL models based on perturbed inputs. Therefore, safeguarding models against these attacks is still an open challenge.

Table 3 Comparison of this review against related surveys and their content coverage

Work	Year	Threats	Solutions	DL	NLP tasks	Metrics
Wagner and Eckhoff (2018)	2018	✓	✓			✓
Sun et al. (2019)	2019	✓	✓	✓	Specific	
Gröndahl and Asokan (2019)	2019	✓	✓	✓	General	
Humbert et al. (2019)	2019	✓	✓	✓		
Boulemtafes et al. (2020)	2020	✓	✓	✓		✓
Kaloudi and Li (2020)	2020	✓		✓	General	
Zhang et al. (2020)	2020	✓		✓	General	✓
This review	2022	✓	✓	✓	General	✓

Unlike the related surveys above, this work covers a broader range of topics to review risks and solutions, especially PETs, for the preservation of privacy in the NLP field. This is highlighted in Table 3, which compares this review against its counterparts in the literature on privacy-preserving DL and NLP. For instance, subjects related to solutions or DL were not approached by some of the works in the table. Therefore, our work covers a broad range of threats, PETs, DL models, NLP tasks, and privacy metrics, bringing a holistic view of privacy preservation for NLP applications.

1.2 Objectives and contributions

Text data can hold private content explicitly, as a user's ID, location, or many demographic attributes, or implicitly as information inferred from the text, like a user's political view (Coavoux et al. 2018). Furthermore, privacy has attracted great attention for developing DL methods for NLP tasks in recent years (Huang et al. 2020; Zhu et al. 2020). Despite the relevance of this topic, there is no extensive review paper that focuses exclusively on privacy-preserving NLP, covering a large number of tasks and PETs. Therefore, this work aims at providing an overview of recent privacy-preserving DL methods for NLP, shaping the landscape of privacy challenges and solutions in this field. We cover all steps of the NLP pipeline, from dataset pre-processing to model evaluation, to come up with a holistic view of privacy issues and solutions for text data. Such a review is needed to help successive scientists and practitioners in the industry have a starting point for the research in privacy-preserving NLP.

The major contribution of this work regards a taxonomy for classifying the existing works in the literature of privacy-preserving NLP, bearing in mind the target for privacy preservation, PETs, the NLP task itself, and the computation scenario. This taxonomy can be easily extended to aggregate future approaches and incorporate new categories of methods. Additional contributions of this review are summarized as follows.

- First, we bring a review of PETs and point out the directions for their efficient integration into NLP models.
- Second, we describe several threats that put the privacy of text data at risk. To provide defenses against these threats, a model has to meet functional requirements related to data types and PETs. Thus, this review helps ease the efforts to find these requirements.
- Third, we introduce and discuss the open challenges to developing privacy-preserving NLP models, taking into account five criteria: traceability, computation overhead, dataset size, bias prevalence in embedding models, and privacy-utility tradeoffs. These criteria affect the suitability of PETs for real-world scenarios.
- Further, we bring an extensive list of benchmark datasets for privacy-preserving NLP tasks so that interested researchers can easily find out baselines for their works.
- Finally, we list metrics to measure the extent privacy can be protected and evaluated in NLP.

1.3 Paper structure

The remainder of this paper is organized as follows. Section 2 outlines the methods for searching and selecting works for this review. Section 3 overviews the theoretical foundations of DL, NLP, and privacy preservation. Section 4 introduces a taxonomy to categorize privacy-preserving NLP methods. Section 5 gives a summary of applications and datasets for privacy-preserving NLP. Section 6 lists metrics for assessing privacy in the NLP field. Section 7 discusses the findings of the review and presents open problems for successive research. Last, Section 8 brings the concluding remarks.

2 Research Method

This review of DL methods for privacy-preserving NLP follows a systematic literature review methodology. We follow the procedure proposed by Kitchenham (2004) in order to retrieve research papers from the existing literature, select relevant works out of the results, and summarize them afterward. Therefore, the systematic review process is reproducible and mitigates selection biases toward the works in the literature. Sections 2.1, 2.2, and 2.3 outline research questions, the search strategy, and the study selection for the making of this review.

2.1 Research questions

Research questions are the cornerstone of a literature review since every step of the review process relies on them (Kitchenham 2004). To come up with this review, we answer the following research question: What are the current DL methods for privacy-preserving NLP, which provide solutions against privacy attacks and threats arising from DL models, computation scenarios, and pieces of private information in text data, such as a person's full name, demographic attributes, health status, and location? For completeness and broader coverage of the privacy-preserving NLP topic, we split the main research question into a row of sub-questions as follows.

- Which PETs have recently been implemented along DL for NLP?
- How can the literature on privacy-preserving NLP be organized into a taxonomy which categorizes similar approaches based on the type of data, NLP task, DL model, and PET?

- How can each PET influence the performance of an NLP model?
- How to select the most suitable PET for an NLP task?
- What are the tradeoffs between privacy preservation and performance for utility tasks in NLP?
- Which privacy metrics can be used for evaluating privacy-preserving NLP models?
- Which benchmark datasets are available for privacy-preserving NLP applications?
- What are the open challenges regarding privacy preservation in the NLP domain?

2.2 Search strategy

Table 4 Privacy and NLP terms used to create search expressions

Term 1	Term 2	NLP Terms
Concealment	Algorithm	Computational linguistics
Confidentiality	Approach	Natural language processing
Privacy	Concept	NLP
Private	Framework	Text analytics
Retreat	Hazard	Text mining
	Idea	
	Manner	
	Means	
	Menace	
	Method	
	Mode	
	Model	
	Path	
	Peril	
	Preservation/Preserving	
	Procedure	
	Protocol	
	Risk	
	Scheme	
	Solution	
	Strategy	
	Technique	
	Threat	
	Way	

The works we review in this article were retrieved from top venues for NLP, machine learning, AI, data security, and privacy, which are indexed by either ACL Anthology[1], ACM Digital Library[2], IEEE Xplore[3], ScienceDirect[4], Scopus[5], SpringerLink[6], or Web of Science[7]. In addition to papers indexed by the aforementioned electronic scientific libraries, we have also included valuable works in e-Print's archive[8] since this repository stores the most up-to-date research results (Zhang et al. 2020). Thus, once the list of libraries was defined, we came up with search terms derived from the main research question to create search strings (Kitchenham 2004). Such strings have been used to conduct the searches on the electronic scientific libraries and retrieve published works afterward.

Table 4 lists the search terms we derived from the main research question in three columns. Firstly, column "Term 1" holds the privacy-related terms. Secondly, column "Term 2" encompasses terms that suggest how privacy is ap-

[1] https://www.aclweb.org/anthology/.
[2] https://dl.acm.org/.
[3] https://ieeexplore.ieee.org/Xplore/home.jsp.
[4] https://www.sciencedirect.com/.
[5] https://scopus.com/search/.
[6] https://link.springer.com/.
[7] https://apps.webofknowledge.com/.
[8] https://arxiv.org/.

proached. Finally, column “NLP Terms” contains NLP-related terms. In total, the table lists 34 different terms which were combined to create search strings for the electronic scientific libraries (Section 2.2.1).

2.2.1 The use of “OR” and “AND” Boolean operators

In order to construct sophisticated search strings, we have combined the terms listed in Table 4 using “OR” and “AND” Boolean operators. First, we selected each term from column “Term 1” and placed it alongside each term from column “Term 2”. Second, we repeated this same step but used the plural form of the term from “Term 2” to replace the original one. For instance, we replaced “risk” with “risks”, “threat” with “threats”, and so forth. Third, we combined the outcomes from the past two steps using the “OR” Boolean operator to come up with the first half of each search string, such as “((“privacy risk” OR “privacy risks”))”. Similarly, we joined all terms from “NLP Terms” using the “OR” Boolean operator to construct the second half for all search strings, as “(((“natural language processing”) OR (“NLP”)) OR ((“text mining”) OR (“text analytics”) OR (“computational linguistics”)))”. Finally, we coupled both search string halves using the “AND” Boolean operator and came up with 120 different search strings to guarantee a wide-stretching coverage of the privacy-preserving NLP literature during the search step.

2.3 Study selection

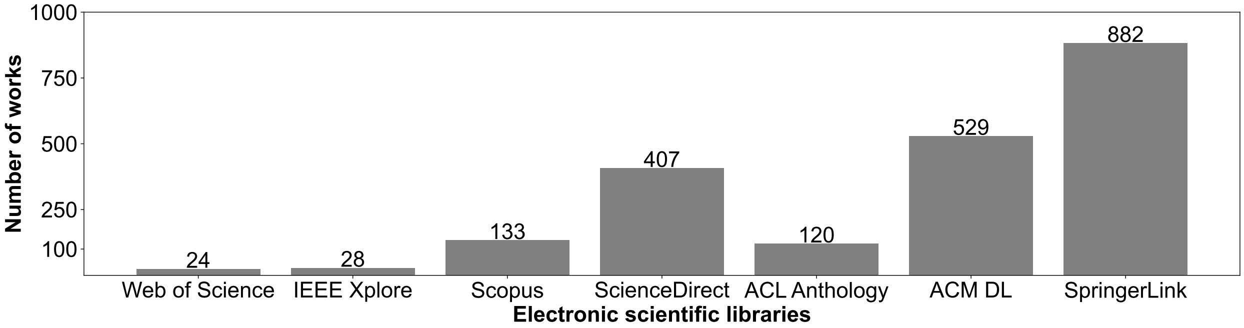


Figure 2 Literature search results

The literature search on the electronic scientific libraries took place after constructing the search strings, as detailed in Section 2.2.1. So we applied each of the 120 search strings on the electronic libraries, retrieving 2,123 works in total. Figure 2 shows the number of works collected from each electronic library at the end of the searches. Searches on SpringerLink, ACM DL, and ScienceDirect returned most of the results, which account for 1,818 papers combined. Therefore, we needed to apply some inclusion and exclusion criteria to select the most relevant works out of this plethora of results.

Published papers that satisfied all the following inclusion criteria were selected for this review.

- I1. Works which employed at least one neural network model in the experimental evaluation, such as CNNs (LeCun et al. 1990), RNNs, BERT (Devlin et al. 2019). For the sake of more extensive coverage, we also included works that reported the use of word embedding models, such as word2vec (Mikolov et al. 2013), GloVe (Pennington et al. 2014), and fasttext (Joulin et al. 2017), since these models are broadly applied to NLP tasks and the privacy threats they may face are similar to those faced by DL architectures.
- I2. Works published from the year 2016 onward. Since this review aims at bringing the most recent developments of privacy-preserving NLP models based on DL, we limited the time range for publications from the past five years.
- I3. Long papers which report the development of privacy-preserving NLP models. These works were preferred over short papers since the surveyed papers were expected to present the complete results for their proposed approaches. However, short papers which demonstrated high impact given the number of citations were also selected.
- I4. Works published by top-tier venues. Many of the papers we review were published at renowned NLP, DL, and privacy conferences, such as ACL, NAACL, EACL, EMNLP, ACM SIGIR, ACM SIGKDD, NEURIPS, IEEE ICDM, and USENIX Conference, or journals, as IEEE Transactions on Pattern Analysis and Machine Intelligence and IEEE Transactions on Information Forensics and Security.

To select e-Prints for this review, we followed three criteria applied by Zhang et al. (2020): paper quality, method novelty, and the number of citations. However, we used an additional criterion, namely publication date. If an e-Print was published before 2018, we have applied a citation number threshold of forty citations. Otherwise, e-Prints published since 2018 were selected if they presented novel and promising approaches for privacy-preserving NLP.

Published works that satisfied any of the following exclusion criteria were removed from this review.

- E.1. Published works that did not report the use of neural network models
- E.2. Works that did not focus on NLP or text data privacy.
- E.3. Works whose datasets used for the experiments did not include text data.
- E.4. Works published before 2016.
- E.5. Duplicated works.
- E.6. Works which consisted of either title page or abstract only.

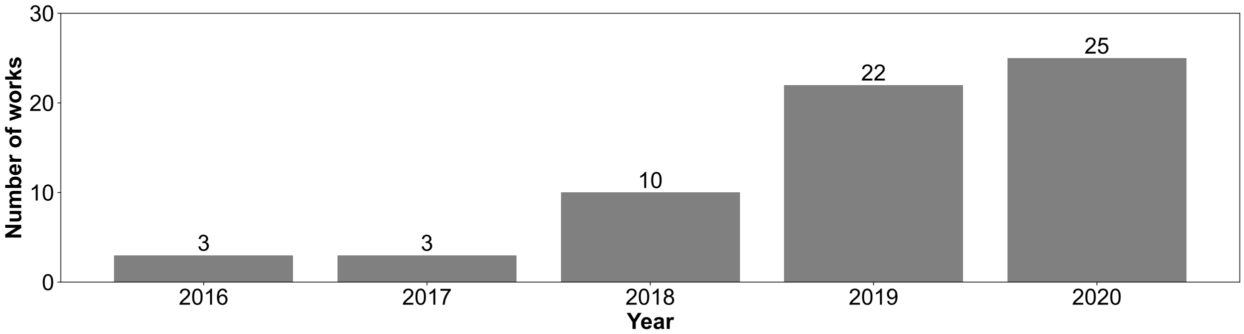


Figure 3 Publication years of the selected works

The selection of works from the literature on privacy-preserving NLP took place between January and February 2021. Consequently, published works indexed by the electronic scientific libraries after February 2021 were not included in this review. After applying the criteria for inclusion and exclusion of works, 63 papers remained to be reviewed. Figure 3 shows the number of papers selected for the review by year from 2016 to 2020. In the figure, it is possible to notice that most of the selected works were published from 2019 onward. In contrast, the number of selected papers published in 2016 and 2017 corresponds to fewer than 10% of the total number of selected works. These results demonstrate the increased interest in privacy for the NLP domain in the past two years. Therefore, this review and its main research question are justified by this ever-increasing interest.

3 Background

Prior to introducing the taxonomy of DL methods for privacy-preserving NLP, we need to present the theoretical background of DL and privacy for NLP. Firstly, we briefly overview DL and its applications for NLP in Section 3.1. Secondly, we introduce the terminology related to privacy in the reviewed papers in Section 3.2. Finally, we list the privacy issues for NLP in Section 3.3.

3.1 Deep learning for natural language processing

DL is a field of machine learning that typically includes neural network architectures featuring multiple layers, a vast number of parameters, and the ability to learn data representations whose abstraction levels are manifold (LeCun et al. 2015). These neural network models can receive enormous amounts of data as inputs and perform inference with high generalization power to reach outstanding performance results (Neyshabur et al. 2017). The training step of a deep neural network encompasses two phases. Firstly, a forward pass over the data is performed (Boulemtafes et al. 2020). Each network layer is initialized with random weights, so bias signals and activation functions, like ReLU (Nair and Hinton 2010), are computed. Then, the model outputs labels for the input data instances in case a supervised task is envisaged. The predicted labels are compared against the ground truth afterward (Boulemtafes et al. 2020). A loss function, as binary cross-entropy (Goodfellow et al. 2016), computes the error rate, which will be passed backward through the network layers, consequently updating their weights and navigating downward the error gradient (LeCun et al. 2015; Boulemtafes et al. 2020). The inference step occurs after the architecture finds a minimum value for the error rate. For this reason, only the forward pass is computed for the testing data, predicting the labels (Boulemtafes et al. 2020). DL architectures can also be trained following unsupervised and semi-supervised settings, such as autoencoders (Hinton and Salakhutdinov 2006) and ladder networks (Rasmus et al. 2015).

In the NLP domain, deep neural networks have brought groundbreaking results to many tasks in the past few years (Feng et al. 2020; Vaswani et al. 2018; Duarte et al. 2021; Sarikaya et al. 2014; Liu et al. 2017a; Saeidi et al. 2019). Text data has demanded the creation of methods specially designed for its representation, such as word embedding and general-purpose language models. We briefly introduce the main DL architectures used in the reviewed privacy-preserving NLP works, such as CNNs and RNNs, and word embedding models, which are frequently implemented alongside DL architectures. The DL models we describe are intrinsically linked to the content of this survey. However, we assume the reader has prior knowledge of such models. Therefore, we omit detailed technical aspects and recommend the reader refer to seminal articles cited next to the architecture's names for complete information if needed.

Convolutional neural networks CNNs (LeCun et al. 1990) are a family of DL architectures originally designed for computer vision tasks and applicable to data types other than images, such as text and tabular data. These models learn feature maps from complex and multi-dimensional inputs using multiple stacked network layers (LeCun et al. 1998). A convolution operation for text consists on the application of a filter $\mathcal{F}$ over a window of words $w'_{i:i+j-1}$ with size j for yielding a new feature c_i in the form

$$c_i = f'(\mathcal{F} \cdot w'_{i:i+j-1} + \mathcal{B}), \tag{1}$$

in which f' is a non-linear function and $\mathcal{B}$ is a bias term (Kim 2014). CNNs are computationally efficient methods that require fewer parameters than architectures solely relying on fully-connected layers. Moreover, the convolution operations are easily suitable for parallelization settings.

Recurrent neural networks RNNs are DL architectures that work with sequential data, such as text, DNA sequences, speech, and time series. The outputs of such models depend on previous computations, which are stored in the model's internal memory, hence exploiting current and past inputs to make a prediction for new data instances (Boulemtafes et al. 2020). Long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997) is a popular RNN variant that deals with the vanishing gradient problem triggered by long input sequences (Feng et al. 2020; Boulemtafes et al. 2020; Hochreiter and Schmidhuber 1997). LSTM takes an input sequence $\{x_1, x_2, \ldots, x_{\mathcal{T}}\}$ and transforms it into another sequence $\{y_1, y_2, \ldots, y_{\mathcal{T}}\}$ by a series of layers, which comprise hidden cells and hidden parts of state memory (Feng et al. 2020). Each layer is composed of gates, namely the input gate, forget gate, and output gate, which control the amount of information to be stored or forgotten by the model in the hidden states, update the hidden states, and decide on the final output. Another popular RNN variant is bidirectional LSTM (BiLSTM) (Graves and Schmidhuber 2005), which encompasses two LSTM models for processing sequence data in both forward and backward directions, hence improving the amount of information and the data context for the architecture (Cornegruta et al. 2016).

General-purpose language models In the past few years, a new paradigm to yield language representations has been noticed. The so-called general-purpose language models comprise giant pre-trained language models which are built upon multiple layers of transformer (Vaswani et al. 2017) blocks and feature millions of parameters learned during the pre-training step on billions of sentences (Pan et al. 2020). These models are designed to encode whole sentences as vectors (embeddings) and, after the pre-training step, are publicly released to be optionally adapted (fine-tuned) for NLP tasks. Another outstanding feature of such models regards their ability to learn new tasks from a few data instances. BERT (Devlin et al. 2019) is a general-purpose model which yields language representations by conditioning the context on both sides of target words. It relies on a loss function based on masking some tokens and trying to predict the word id of the masked words solely based on the surrounding context. There are two standard model sizes for BERT concerning the number of layers (L), hidden size ($\mathcal{H}$), self-attention heads ($\mathcal{A}$), and parameters (Θ) used to build the model architectures. The first one is BERT$_{BASE}(L = 12, \mathcal{H} = 768, \mathcal{A} = 12, \Theta = 110M)$, and BERT$_{LARGE}(L = 24, \mathcal{H} = 1024, \mathcal{A} = 16, \Theta = 340M)$ is the second architecture. BERT variants have pushed state-of-the-art results forward in many NLP tasks. However, BERT-based models require a high memory footprint due to the enormous parameter sizes. Additional huge models in the same category as BERT are GPT (Radford et al. 2018), GPT-2 (Radford et al. 2019), and GPT-3 (Brown et al. 2020).

Word embedding models Word embeddings are distributed representations of words yielded by shallow neural networks in order to reduce the number of dimensions of such vector representations, whereas semantic features of words, such as context, are preserved (Camacho-Collados and Pilehvar 2018). These representations are extensively used across NLP tasks, mostly combined with DL architectures. Therefore, based on the relatedness between word embeddings and DL, we also include privacy-related topics arising from embedding models in this review. Popular word embedding models are word2vec (Mikolov et al. 2013), GloVe (Pennington et al. 2014), and fasttext (Joulin et al. 2017). All these models are able to capture semantic properties of words (Camacho-Collados and Pilehvar 2018), hence representing related terms as close coordinates in a vector space.

3.2 Privacy terminology

Table 5 Topics and terminology for privacy preservation for NLP

Topic	Key terms
Bias and fairness in NLP	Demographic attributes, adversarial training, adversarial components, debiasing, gender-neutral word embeddings, approximate fairness risk, attributes obfuscation, fair representation learning, neural machine translation
Privacy-preserving NLP	Private training, private inference, identifiable words, nontransferable data, laws and regulations, private information, lack of trust, non-traceability, data safeguarding, PETs
Disentangled representations	Factors of variation, optimization, separate latent spaces, adversarial objective functions, transfer learning, style transfer

Text data may contain pieces of private information explicitly or implicitly integrated into its content, such as precise key phrases or demographic attributes inferred from the context (Coavoux et al. 2018). Furthermore, privacy-related issues may occur at any phase of the NLP pipeline, from data pre-processing to downstream applications. Privacy preservation, for this reason, involves the perspectives of data (Chen et al. 2019), DL model (Song and Raghunathan 2020), PETs (Melamud and Shivade 2019), fairness (Ekstrand et al. 2018), computation scenario (Feng et al. 2020), downstream tasks (Sousa et al. 2021), and the interplay between these perspectives (Ekstrand et al. 2018). These multiple perspectives contribute to ever-increasing literature from which many privacy-related topics develop. In Table 5, we list the three major topics for privacy preservation for NLP beside their related key terms. First, bias and fairness in NLP is a topic regarding automated decision-making systems which are prone to biases arising from pieces of private information, like gender (Elazar and Goldberg 2018), yet wished not to prompt discriminatory outputs (Ekstrand et al. 2018). The efforts to promote fair decision-making often interact with those for privacy protection (Ekstrand et al. 2018). Second, privacy-preserving NLP is a catchphrase for approaches that protect privacy for models combined with PETs (Feng et al. 2020; Alawad et al. 2020). Finally, disentangled representations play a key role in integrating privacy premises into learning data representations used for downstream tasks (Lee and Pavlovic 2021). We further describe each of these topics in the paragraphs below.

Bias in machine learning is a concept related to unfair or discriminatory decisions made by models towards or against specific individuals or groups (Ntoutsi et al. 2020). It becomes a severe problem in real-world scenarios since those decisions may directly impact a person's life or society as a whole. For instance, gender-biased job advertising tools were found to suggest lower-paying job positions to women more often than men (Datta et al. 2015). Human biases are long-lasting research topics in philosophy, social sciences, psychology, and law (Ntoutsi et al. 2020). There are three types of bias: preexisting bias (from the data), technical bias (from models with computational or mathematical constraints), and emergent bias (from the evaluation of results and their applicability) (Papakyriakopoulos et al. 2020). Recently, this subject has also been gaining attention in the NLP field since word representations, like the widely used word embeddings (Mikolov et al. 2013), can also be under threat of encoding human bias (Kaneko and Bollegala 2019) regarding gender, ethnicity, or social stereotypes from text corpora used for training. Therefore, applications built on such biased NLP models have the potential to amplify such misuses of language and propagate them to inference steps. **Fairness** is then typically associated with trustworthy AI (Floridi 2019) and defined as an assurance against discriminatory decisions by AI models based on sensitive features (Zhang and Bareinboim 2018). Privacy technologies and policies often go hand in hand with concepts of fairness (Ekstrand et al. 2018). Additionally, works approaching bias and fairness in NLP follow the premise that the input data present sensitive features, e.g., gender (Elazar and Goldberg 2018; Zhao et al. 2018), which may lead to unfair predictions by downstream systems or be recovered from representations. Therefore, debiasing and attribute removal are techniques to mitigate the undesired discriminatory effects arising from unfair models. Examples of use case tasks covering bias and fairness encompass learning gender-neutral word embeddings (Zhao et al. 2018; Bolukbasi et al. 2016), analysis and reduction of gender bias in multi-lingual word embeddings (Zhao et al. 2020; Font and Costa-jussà 2019), text rewriting (Xu et al. 2019), analysis of biases in contextualized word representations (Tan and Celis 2019; Hutchinson et al. 2020; Gonen and Goldberg 2019; Basta et al. 2020), detection, reduction and evaluation of biases for demographic attributes in word embeddings (Papakyriakopoulos et al. 2020; Sweeney and Najafian 2020, 2019; Kaneko and Bollegala 2019), analogy detection (Nissim et al. 2020), cyberbullying text detection (Gencoglu 2020), fair representation learning (Friedrich et al. 2019), protected attributes removal (Elazar and Goldberg 2018; Barrett et al. 2019), analysis of racial disparity in NLP (Blodgett and O'Connor 2017), and prediction of scientific papers authorship during double-blind review (Caragea et al. 2019).

Privacy-preserving NLP is an expression that refers to language models trained or used for inference on private data without putting privacy at risk. Some assumptions are considered for the development of such methods: (i) encoded sensitive information about the input must be kept private (Coavoux et al. 2018; Mosallanezhad et al. 2019; Feyisetan et al. 2019) (e.g., personal attributes, demographic features, location, etc.); (ii) the model's vocabulary may contain words that easily identify people in the data (Alawad et al. 2020; Li et al. 2018); (iii) personal data should never leave their owner's devices (Chen et al. 2019; Alawad et al. 2020; Hard et al. 2018); (iv) the data are subject to legal terms and regulations (Clinchant et al. 2016; Melamud and Shivade 2019; Belli et al. 2020; Battaglia et al. 2020; Martinelli et al. 2020); (v) private information may be correlated with the labels of the model outputs with high likelihood and learned thereby (Coavoux et al. 2018; Li et al. 2018; Song and Raghunathan 2020; Carlini et al. 2019); (vi) the computation scenario is not trusted against privacy attacks and threats, such as eavesdropping, breaches, leaks, or disclosures (Feng et al. 2020; Coavoux et al. 2018; Dai et al. 2019; Feyisetan et al. 2020; Liu and Wang 2020); (vii) the input data must be untraceable to any users but the data owner (Oak et al. 2016). Among the PETs for NLP, there are anonymization (Oak et al. 2016), data sanitization (Feyisetan et al. 2019), data obfuscation (Martinelli et al. 2020), text categorization (Battaglia et al. 2020), transfer learning (Alawad et al. 2020; Song and Raghunathan 2020), FL (Chen et al. 2019; Hard et al. 2018), black box model adaptation (Clinchant et al. 2016), encryption (Dai et al. 2019; Liu and Wang 2020), MPC (Feng et al. 2020), DP (Feyisetan et al. 2020; Melamud and Shivade 2019), adversarial learning (Li et al. 2018), deep reinforcement learning (Mosallanezhad et al. 2019), and generative models (Carlini et al. 2019).

Disentangled representations regard the premise that good data representations capture factors of variation from the input feature space and represent these factors separately (Bengio 2009). Latent spaces of neural networks can be, thus, disentangled as to different features, e.g., adding terms to the model's objective functions as an adversarial training setting (John et al. 2019). Images in computer vision tasks are popular targets for learning disentangled representations (Mathieu et al. 2016; Lee and Pavlovic 2021). Moreover, recent NLP approaches helped yield representations for language features, such as style and content, as separate latent spaces for style transfer tasks to be performed afterward (John et al. 2019). Therefore, sensitive features are preserved while the representations for the remaining ones can be used for applications without putting privacy at risk.

3.3 Privacy issues for NLP

Privacy issues arise in situations where an attacker can successfully associate a record owner to a sensitive attribute in a published database (Menzies et al. 2015), disclose model inputs (Li et al. 2018; Song and Raghunathan 2020; Huang et al. 2020), obtain information that should be kept private (Lyu et al. 2020; Coavoux et al. 2018; Feng et al. 2020), among other harmful activities. Consequently, unintended data breaches may occur and lead to problems, such as social exposure, documents leakage, and damages to an individual's or organization's reputation. Furthermore, data protection laws establish penalties and fines if a data breach happens. Therefore, when a DL model is designed to process personal data, it is crucial to consider the privacy threats that put this model at risk of data breaches.

We list over fifteen different privacy threats and attacks to ease their identification for designing DL models for privacy-preserving NLP. We take into account three perspectives to grouping privacy threats in NLP. Firstly, the threats arising from datasets that are made public and therefore can have their original content disclosed. Secondly, the threats related to how DL models can violate data privacy. For instance, a model can memorize protected attributes and allow their disclosure later on (Kumar et al. 2019). Another model threat regards how language models address human discriminatory biases from their training text corpora (Basta et al. 2020; Sweeney and Najafian 2020). Moreover, the computation scenario, such as centralized cloud servers or distributed processing architectures, plays an important role in the existence of privacy threats since many of them are related to the misbehavior of components. So we present the most common threats from the computation scenario in the reviewed papers. Finally, we overview privacy attacks that target DL models for NLP.

3.3.1 Threats from data

From the data perspective, the most common privacy threats for text data are related to its content (Clinchant et al. 2016), which encompasses pieces of private information like identities of authors, health status, sentiment polarities,

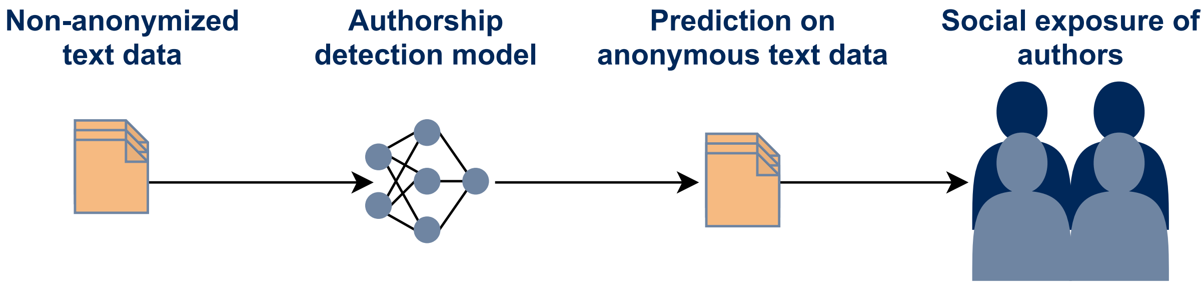

Non-anonymized text data
Authorship detection model
Prediction on anonymous text data
Social exposure of authors

Figure 4 Re-identification of anonymous texts. In this setting, an authorship detection model can threaten the identities of anonymous text data authors from social exposure.

and demographic attributes. In the reviewed works, we identified the following privacy threats arising from the data perspective.

- *Hardness to tag sensitive information*. Data privacy frequently relies on the obfuscation of sensitive information of documents, which is a hard task since all direct and indirect informational clues that may identify a person should be obfuscated (Martinelli et al. 2020). Furthermore, the sensitive content may be expressed by words that are not sensitive themselves, as describing sensitive bank transactions using the same vocabulary as descriptions of non-sensitive ones, but with different natural language expressions (Neerbek et al. 2018). Therefore, information taggers based on keyword lists are susceptible to failure.
- *Re-identification of documents and anonymous text*. Documents such as electronic medical records and government reports are usually publicly released for the sake of leveraging research on their respective domains or government transparency (Fernandes et al. 2019). Although these documents are required to be sanitized by the removal of their authorship information to protect their authors, re-identification by malicious models can still take place (Fernandes et al. 2019). Online anonymous texts, such as e-mails, comments, and blog posts, can also be re-identified by authorship detection models trained on non-anonymized data (Seroussi et al. 2014), as depicted by Figure 4. Additionally, authorship identification models can have beneficial applications to intellectual property management (Boumber et al. 2018), as in plagiarism detection. Thus, formal guarantees against re-identification of documents have to be provided by efficient de-identification methods, for instance, via DP (Dwork 2008; Fernandes et al. 2019).
- *Re-identification of anonymous source code*. Source code of open source projects can be used to identify the developers based on their coding style (Abuhamad et al. 2018, 2019). This threat is particularly dangerous when developers do not wish to expose their identities.
- *Self-disclosure of emotions and personal information*. Social media posts often carry private information voluntarily released by users, such as gender, location, career, and feelings towards things and people (Akiti et al. 2020; Battaglia et al. 2020). These users are frequently not aware of the sensitivity of the information they post so that models trained on such data can be input with private information without notice.

3.3.2 Threats from models

Model properties, such as vocabulary and neural network layers, can be used by an adversary to perform attacks that end up disclosing private data used for training (Alawad et al. 2020). Additional privacy-related issues from NLP models concern biases (Sweeney and Najafian 2020; Gencoglu 2020; Tan and Celis 2019) and the unfair decisions made by biased models (Sweeney and Najafian 2020; Xu et al. 2019). Therefore, from the perspective of NLP models, privacy threats are as follows.

- *Bias in word embedding models*. The encoding, amplification, and propagation of human discriminatory biases are noticeable privacy-related issues for word embedding models. The most common types of encoded biases regard gender (Basta et al. 2020; Bolukbasi et al. 2016; Font and Costa-jussà 2019; Gencoglu 2020; Kaneko and Bollegala 2019; Nissim et al. 2020; Papakyriakopoulos et al. 2020; Sweeney and Najafian 2020; Tan and Celis 2019; Vig et al. 2020; Zhao et al. 2018), race (Sweeney and Najafian 2020; Tan and Celis 2019), professions (Papakyriakopoulos et al. 2020), religion (Sweeney and Najafian 2020), intersectional identities (Tan and Celis 2019), language (Gencoglu 2020), and disabilities (Hutchinson et al. 2020), to name a few. The removal or lessening of such issues is challenging since the semantics of the representations yielded by the models should be preserved, whereas the discriminatory biases should be removed to the largest extent possible. Given the hardness of performing de-biasing of embedding models, biased or unfair decisions towards demographic attributes can be made (Xu et al. 2019; Sweeney and Najafian 2020; Bolukbasi et al. 2016). Furthermore, it was found that bias may present some prevalence on debiased embedding models (Gonen and Goldberg 2019), hardening its complete removal. Finally, the transfer of gender bias across languages in multilingual word embeddings is another threat to be taken into account (Zhao et al. 2020). For instance, word embeddings generated for a neural machine translation task (Feng et al. 2020), which translates sentences from Spanish into English, may capture gender-related bias from Spanish and integrate it into the embeddings of English words.

- *Disclosure of protected health information*. Patient data is inherently private since it holds attributes related to a person's identity, health status, diagnosis, medication, and demographic information. It is protected by regulations, such as the Health Insurance Portability and Accountability Act (HIPAA) (Act 1996) in the United States, hence demanding de-identification prior to the public release of such data for research activities (Liu et al. 2017b; Dernoncourt et al. 2017; Obeid et al. 2019). NLP models for healthcare data often suffer threats from sharing vocabulary dictionaries, which encompass entries related to patient identities, for the embedding layer of neural networks (Alawad et al. 2020).

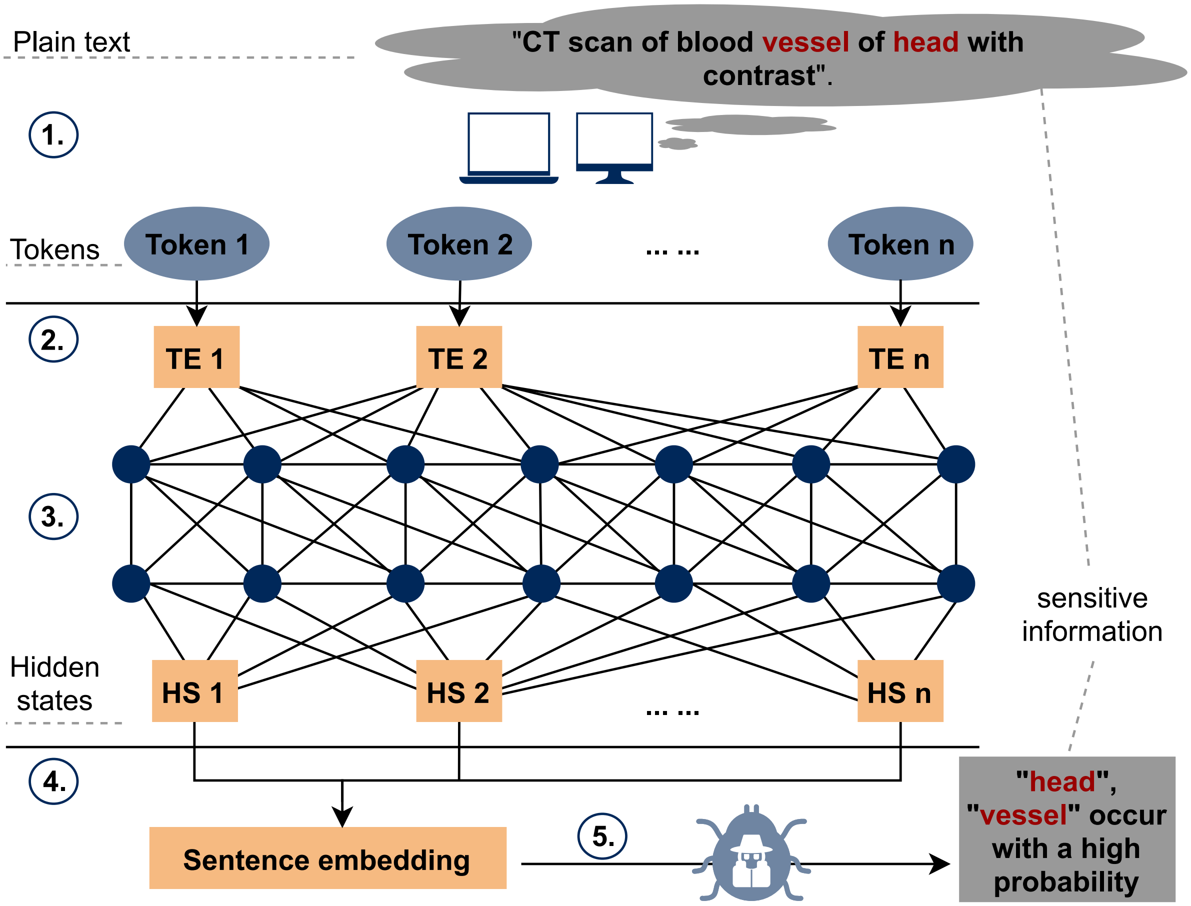


Figure 5 Unintended feature memorization (Pan et al. 2020). The bold red text represents privacy-sensitive features memorized by the general-purpose language model. The unintended memorization and successive reconstruction of such features from sentence embeddings can occur as follows. **1.** Plain text tokenization. **2.** Token embedding (TE). **3.** Propagation across transformer layers. **4.** Pooling over hidden states (HS) to yield a sentence embedding. **5.** Inference of sensitive information from sentence embeddings.

- *Unintended feature memorization*. Firstly, deep neural NLP models can learn topical features containing spurious correlations from the training data, which damage the model performance on prediction tasks (Kumar et al. 2019). For instance, the prediction of an author's native language from texts written in a second language can be biased towards geographical terms frequently occurring in their text, like country names, yet not related to the task target. Consequently, these models are likely to fail to generalize on new data instances because of such learned features (Kumar et al. 2019), which may also include pieces of private information. Secondly, general-purpose language models with millions of learnable parameters, like BERT and GPT, also capture sensitive information from the training data, hence being at risk from model inversion attacks, which retrieve this information from the trained model (Pan et al. 2020). Figure 5 depicts the unintended memorization of privacy-sensitive attributes from the text while yielding sentence embeddings. In the figure, an attacker infers these attributes from the sentence embeddings afterward. Finally, DL models were also found to memorize unique or rare training data sequences (Carlini et al. 2019). Therefore, first-line defenses like removing such pieces of private information can hinder this issue.

3.3.3 Threats from the computation scenario

The computation scenario plays an important role in putting privacy at risk. For instance, communication channels, servers (Dai et al. 2019), or computation parties (Feng et al. 2020) may behave in unreliable manners. Therefore, the following threats can take place.

- *Disclosure of data from user devices*. In FL scenarios, model parameters may allow adversaries to learn about the original training data (McMahan et al. 2018). For instance, model gradients computed locally on users' devices can carry implicit private information from the locally stored data like user's behavior (Qi et al. 2020). Therefore, DP methods can be used to provide privacy guarantees at the user level (McMahan et al. 2018).
- *Honest-but-curious server*. Another example of a semi-honest security model regards cloud or central servers that are not fully trustworthy (Dai et al. 2019; Zhu et al. 2020). This threat requires encryption or DP noise to be applied to the data or model parameters before outsourcing to the server.
- *Semi-honest security model*. This threat, also referred to as the honest-but-curious security model, frequently happens in MPC settings, in which a corrupted party follows the MPC protocols exactly, but this party tries to learn more information than expected during the iterations (Feng et al. 2020).

3.3.4 Privacy attacks

In DL-based NLP, privacy attacks aim at leaking data samples used for model training, exposing individuals and their private information, such as identity, gender, and location. Many factors influence the likelihood of success an attacker model can obtain, e.g., the DL model itself (Mosallanezhad et al. 2019), the computation scenario (Lyu et al. 2020), and the data properties (Oak et al. 2016). Therefore, there is a large number of privacy attacks that target text data used to train DL models. In the surveyed works, we identified nine different privacy attacks, which we describe as follows.

- *Adversarial attacks*. Malicious modifications of texts with a small impact on the readability by humans, yet able to make DL models output wrong labels, constitute adversarial attacks (Liu and Wang 2020; Zhang et al. 2020). Examples of such modifications include character-level perturbations and the replacement of words by semantic similarity or probability.
- *Membership inference attacks*. Membership inference attacks comprise a widely researched class of attacks against DL models. In these attacks, an adversary attempts to disclose the 'is-in' relation between a data sample and the original private training set of a model (Pan et al. 2020).
- *Attribute inference attacks*. Text data encompasses a large number of private attributes which can be leaked through embeddings for words, sentences, or texts, which are trained without efficient anonymization. These attributes include gender, age, location, political views, and sexual orientation (Mosallanezhad et al. 2019). For instance, this attack can take place as a classifier that predicts private information, like location, of real-time system users from the embeddings of their texts (Elazar and Goldberg 2018; Mosallanezhad et al. 2019; Barrett et al. 2019). Thus, language representations shared over different tasks or computation parties have to be robust against those attacks.
- *Re-identification attacks*. Anonymized documents can still hold information that can be used to trace back the individuals who generated it, using auxiliary data sources (Oak et al. 2016). An example of such an attack was the re-identification of the Netflix Prize dataset, a database composed of 100,480,507 movie ratings of 480,189 Netflix users in the years between 1999 and 2005, using the IMDB dataset as a surrogate for underlying knowledge about the attack targets (Narayanan and Shmatikov 2008).

- *Eavesdropping attacks*. An eavesdropping attack happens in scenarios in which the computation is distributed across many devices (Lyu et al. 2020), e.g., FL. Thus, one of the devices would try to infer private information from, for instance, latent representations sent to a cloud server by other devices in the setting (Coavoux et al. 2018).
- *File injection attacks*. Searchable encryption (Cash et al. 2015) can have its privacy guarantees broken by file injection attacks, which can be seen as a more general class of adversarial attack. In such attacks, an adversary injects files composed of keywords into a client of a cloud server, which will encrypt the injected files and store them on the server (Liu and Wang 2020). Therefore, the attacker will observe the patterns of the encrypted files, threatening query files, and disclose user keywords (Liu and Wang 2020).
- *Reverse engineering attacks for language models*. DL models for NLP can be easily reverted by an adversary that has prior knowledge of the model (Li et al. 2018). Consequently, this adversary may be able to reverse engineer the input data sampled and leak private information from the training examples. Embedding models are prone to these attacks since word vectors also leak information about the input data (Song and Raghunathan 2020). Therefore, preventing reverse engineering attacks for NLP is a tricky challenge, especially for FL settings, in which the training cannot be slowed down, and the model accuracy should fall short. In FL scenarios, the adversaries can be corrupted devices that have access to information communicated by all parties in the computation like parameters of the model during the training (Huang et al. 2020). Potential solutions include DP (Dwork 2008), FHE (Gentry 2009), or the combination of both (Huang et al. 2020).
- *Pattern reconstruction attacks*. For pattern reconstruction attacks, the text in its original format presents a fixed structure, like a genome sequence, and the adversary tries to recover a specific part of this sequence that contains a piece of sensitive information (Pan et al. 2020). A gene expression related to an illness can be an example of such sensitive information.
- *Keyword inference attacks*. Sometimes the adversary is solely interested in probing whether a plaintext includes a given sensitive keyword (Pan et al. 2020). For instance, the plaintext can be a clinical note, and the sensitive keyword can be a disease location. Therefore, the adversary tries to recover her/his interested keywords.
- *Property inference attacks*. Unlike membership inference attacks, property inference attacks regard the attempts of an adversary to discover global properties of the original training set, such as the class distribution (Pan et al. 2020). These attacks pose privacy threats for NLP models since the predicted properties may not be shared by the model producer in a consenting manner (Ganju et al. 2018). For instance, an adversary may intend to disclose the male-female ratio of a given population related to the electors of a political party. In order to mask this sensitive information against property inference attacks, a model can apply DP noise to the dataset.

4 Deep learning methods for privacy-preserving NLP

The protection of privacy in NLP is a challenge whose solution depends on many factors, such as computation scenario, utility performance, memory footprint, dataset size, data properties, NLP task, and DL model. Consequently, choosing a suitable PET is not a problem of solely protecting the greatest extent of privacy as possible since privacy-utility tradeoffs can either turn a solution feasible for a real-world application or impractical otherwise. In the past few years, privacy has been attracting significant attention. So many works have been addressing it for DL and NLP, yet making it hard to follow the progress of the literature due to lack of categorization. Therefore, we propose a taxonomy that organizes this literature and shapes the landscape of DL methods for privacy-preserving NLP.

4.1 Categories of DL methods for privacy-preserving NLP

When it comes to privacy-preserving NLP approaches based on DL, we can find similarities between methods considering two major factors: the target of privacy preservation and the PETs specifically. The former determines where privacy is assured, such as in the dataset prior to training and inference, model components during the learning phase, or post-processing routines. The latter specifies which existing PETs are appropriate for each privacy scenario. For instance, encryption is recommended when the server where the data is stored, or another computation party, is no longer trusted. So we gather the methods which implement encryption schemes for utility tasks of NLP into a group of encryption methods. Additionally, since encryption methods are commonly implemented alongside a DL model and remain in place during model training and inference, we insert them into the category of methods whose privacy focus is on the model side, namely trusted methods. This category is divided into two sub-categories according to the computation scenarios for which the trusted methods are implemented. Similarly, we followed this insight to come

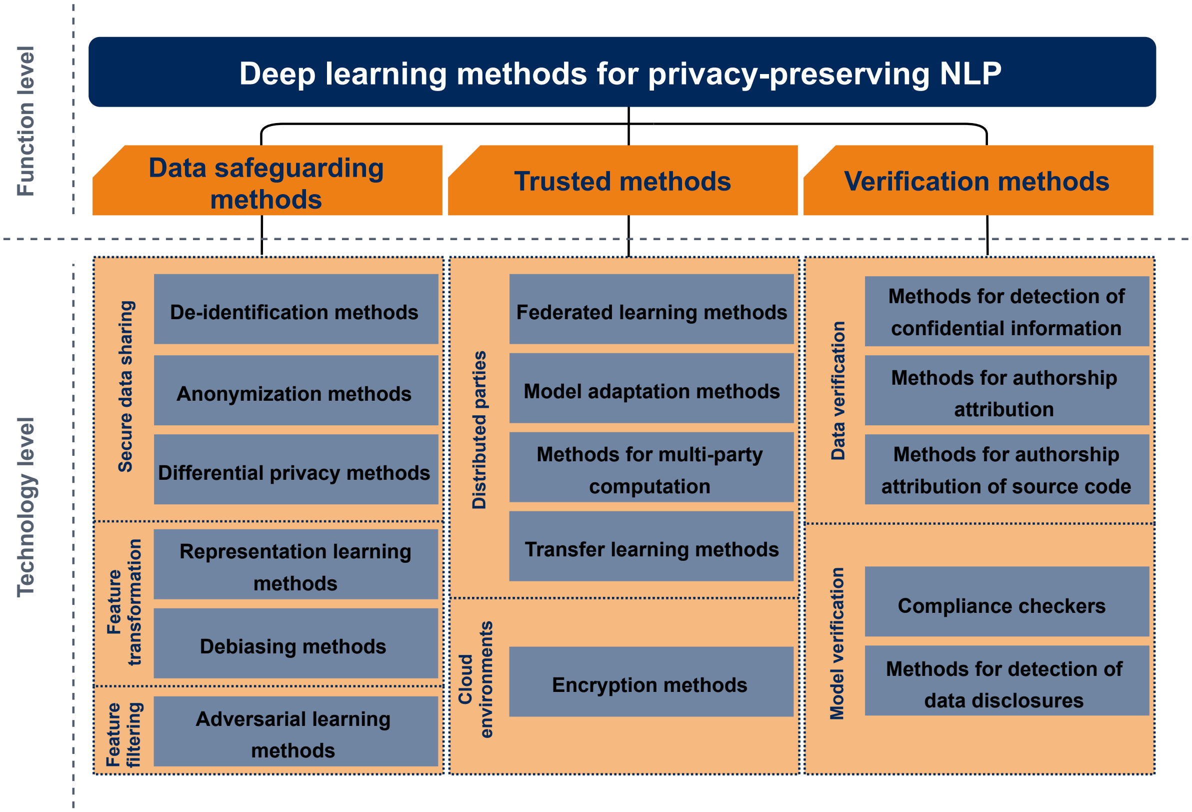

Function level
Technology level
Deep learning methods for privacy-preserving NLP
Data safeguarding methods
Trusted methods
Verification methods
Secure data sharing
De-identification methods
Anonymization methods
Differential privacy methods
Feature transformation
Representation learning methods
Debiasing methods
Feature filtering
Adversarial learning methods
Distributed parties
Federated learning methods
Model adaptation methods
Methods for multi-party computation
Transfer learning methods
Cloud environments
Encryption methods
Data verification
Methods for detection of confidential information
Methods for authorship attribution
Methods for authorship attribution of source code
Model verification
Compliance checkers
Methods for detection of data disclosures

Figure 6 Taxonomy of DL methods for privacy-preserving NLP

up with a taxonomy (Figure 6) which is composed of two levels, three categories, seven sub-categories, and sixteen groups for the surveyed methods and their respective PETs.

In Figure 6, we depict the full proposed taxonomy in two levels: function level and technology level. The former refers to the target of privacy preservation throughout the NLP pipeline, and the latter separates the methods into groups based on the PETs they implement and the computation scenarios they approach. In the taxonomy structure, there are three categories of methods. First, the category of data safeguarding methods is the most extensive in the structure. It aggregates methods divided into six groups, accounting for twenty-seven works in total, covering PETs that are run before the model training and inference, such as debiasing of word embeddings. Second, methods for privacy preservation during model training or inference constitute the category of trusted methods. This category encompasses five groups which account for fourteen works in total. Finally, the category of verification methods includes the remaining five groups with twenty-two works, which aim to detect confidential information in text documents or even assess how susceptible to privacy threats DL models for privacy-preserving NLP are. This taxonomy serves as a framework for categorizing the existing literature on privacy-preserving NLP, easily extendable for aggregating successive works. Moreover, it helps researchers and practitioners identify the most suitable PET for their needs.

4.2 Data safeguarding methods

Data safeguarding methods are applied over datasets shared between NLP tasks or used for downstream applications. Figure 7 depicts these groups, which approach secure data sharing, feature transformation, and feature filtering. Subsequently, Table 6 summarizes the groups of works for data safeguarding, including their neural network models, PETs, and computation scenarios.

4.2.1 De-identification methods

De-identification of data instances can be seen as a named-entity recognition problem (Liu et al. 2017b) to scrub private content from the text by detecting and replacing it with synthetic instances of text, mostly from domains that deal with protected health information (PHI) (El Emam et al. 2009). For instance, a patient's name can be supplanted by a generic

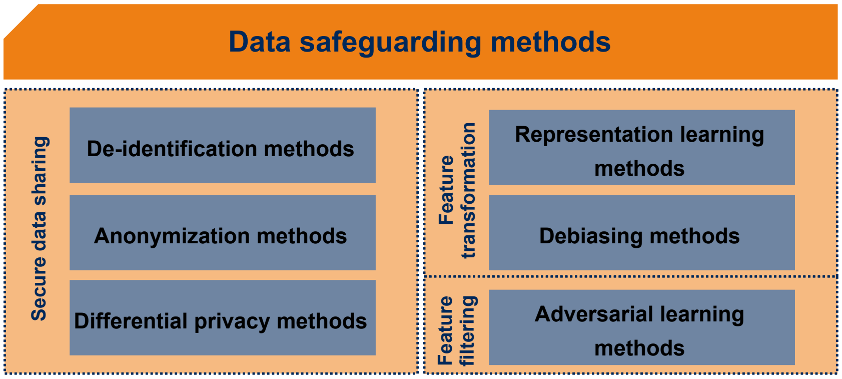

Data safeguarding methods
Secure data sharing
De-identification methods
Anonymization methods
Differential privacy methods
Feature transformation
Representation learning methods
Debiasing methods
Feature filtering
Adversarial learning methods

Figure 7 Sub-categories and groups for data safeguarding methods

tag (e.g., 'PHI'), a private information class descriptor (e.g., 'patient'), or a randomly generated surrogate word from the same information class (e.g., a pseudonym) (Obeid et al. 2019; Eder et al. 2020). Furthermore, regulations like the HIPAA (Act 1996) in the United States enforce any PHI to be safeguarded from disclosures without the patient's permission or awareness. However, given the number of medical records generated every day, the de-identification task demands methods that are able to handle big databases. Liu et al. (2017b) develop an ensemble model for this task using three different methods combined with a regular expression-based sub-system to identify PHI. These methods were a conditional random field and two BiLSTMs. Hence, each method was able to identify PHI unidentified by the other two. Similarly, Dernoncourt et al. (2017) introduce a de-identification system for medical records based on a BiLSTM architecture without relying on handcrafted features. Finally, Obeid et al. (2019) analyze how de-identification of clinical text impacts the performance of machine learning and DL models. The authors take into account the tradeoff between privacy protection and electronic health records utility degradation for tasks of machine learning and information extraction by comparing the performance of machine learning classifiers and CNN models on both original and de-identified versions of a medical dataset. In this scenario, PHI attributes, such as dates, may improve classification outcomes but simultaneously breach private information of patients. Thus, the de-identification process gets rid of these attributes in order to comply with data protection regulations.

Another source of private information that should be de-identified to safeguard privacy is user-generated content, such as e-mails, social media posts, online comments, chats, and SMS. This content poses privacy threats to people, locations, and entities described in the piece of text besides its writer. Eder et al. (2020) investigate the automatic recognition of privacy-sensitive attributes in e-mails and come up with a solution for privacy protection using BiLSTMs and sub-word embedding models for recognition and pseudonymization of identifying information of people in the messages. The authors conducted their experimental evaluation on two e-mail datasets for the German language, starting with the manual annotation of named entities related to attributes to be protected. Further, four neural models for the recognition of private entities are tested and benchmarked. Finally, the private entities are replaced with pseudonym forms that preserve the information type. Therefore, at the end of the experiments, the authors also provided a pseudonymized German e-mail corpus for additional research.

The protection of user-generated text also influences research areas that use examples directly from data to bring examples of key findings, such as behavioral research, since those examples can lead to the re-identification of private data. Oak et al. (2016) produce synthetic data from users' discourse about life-changing events on social media. To do so, the authors use an LSTM model fed with tweets concerning the events of birth, death, marriage, and divorce. The model is first trained to predict the most probable data item (character or word) given a data item used for input and, finally, used for language generation. The synthetic data generated by the model are compared against real tweets held out from training for the evaluation step. Subsequently, human annotators indicate if they thought a tweet was human- or machine-generated. Besides yielding realistic-looking text with similar statistical features as the training data, the authors bring insights on downstream applications, e.g., the utility for software developers who dismiss access to data in raw form for software development activities.

4.2.2 Anonymization methods

In privacy-preserving data publishing, the data must be anonymized prior to its release, aiming at averting attacks that put privacy at risk (Zhou et al. 2008). Data anonymization consists of the removal of all pieces of sensitive

Table 6 Summary of data safeguarding methods

Group	Work	Neural models	PET	*T*	*S*
DI methods	Liu et al. (2017b)	BiLSTM	RE	*A*	MR
	Dernoncourt et al. (2017)	GloVe, word2vec, BiLSTM	PHI detection	*A*	MR
	Obeid et al. (2019)	Word2vec, CNN	BoB	*D*	MR
	Eder et al. (2020)	GERMANER, NEURONER, GERMAN NER, BPEMB	EAR	*A*	E-mails
	Oak et al. (2016)	LSTM	EAR	*A*	Tweets
AM	Mosallanezhad et al. (2019)	GloVe, BiLSTM + attention	DRL	*A*	PA
	Sánchez et al. (2018)	CNN	SA	*D*	PA
	Pablos et al. (2020)	BERT	EAR	*D*	CD
DP methods	Feyisetan et al. (2020)	GloVe, fasttext, BiLSTM	$d_{\mathcal{X}}$-privacy	*W*	PA
	Fernandes et al. (2019)	Word2vec, fasttext	$d_{\mathcal{X}}$-privacy	*A*	PA
	Melamud and Shivade (2019)	Word2vec, LSTM	S-PDTP	*A*	CN
	Lyu et al. (2020)	BERT, MLP	ϵ-DP	*W*	Cloud
RL methods	Li et al. (2018)	BiLSTM, word2vec, CNN	AL	*A*	PA
	Feyisetan et al. (2019)	GloVe, SkipThought, Fasttext, InferSent	$d_{\mathcal{X}}$-privacy	*W*	PA
	John et al. (2019)	DAE, VAE, word2vec, CNN	Auxiliary losses	*A*	ST

Continues on next page...

Table 6 Summary of data safeguarding methods (continued)

Group	**Work**	**Neural models**	**PET**	***T***	***S***
Debiasing methods	Bolukbasi et al. (2016)	Word2vec	Debiasing	*W*	TL
	Kaneko and Bollegala (2019)	GloVe, Hard-GloVe, GN-GloVe, autoencoders	Debiasing	*W*	TL
	Font and Costa-jussà (2019)	GloVe, GN-GloVe, transformer Hard-Debiased GN-GloVe	Debiasing	*W*	NMT
	Papakyriakopoulos et al. (2020)	GloVe, LSTM	VST	*W*	SM
	Gencoglu (2020)	sentence-DistilBERT	FC	*W*	OT
AL methods	Friedrich et al. (2019)	Fasttext, GloVe, BiLSTM-CRF	RL, DIM	*R*	MR
	Elazar and Goldberg (2018)	LSTM, MLP	AT	*A*	PA
	Barrett et al. (2019)	LSTM, MLP	AT	*A*	PA
	Xu et al. (2019)	Transformer	RL, AT	*A*	PA
	Coavoux et al. (2018)	LSTM	RL, AT	*A*	PA
	Kumar et al. (2019)	BiLSTM + attention	RL, AT	*W*	PA
	Sweeney and Najafian (2020)	Word2vec, GloVe, LSTM, CNN	RL, AT	*A*	PA

A stands for a set of protected attributes, **AL** stands for ‘adversarial learning’, **AM** stands for ‘anonymization methods’, **AT** stands for ‘adversarial training’, **BoB** stands for ‘best-of-breed clinical text de-identification application’ (Ferrández et al. 2013), **CD** stands for ‘clinical data’, **CN** stands for ‘clinical notes’, **CRF** stands for ‘conditional random field’, ***D*** stands for a set of documents, **DAE** stands for ‘deterministic autoencoder’, **DI** stands for ‘de-identification’, **DIM** stands for ‘de-identification model’, **DP** stands for ‘differential privacy’, **DRL** stands for ‘deep reinforcement learning’, **EAR** stands for ‘entity annotation and recognition’, **FC** stands for ‘fairness constraints’, **MR** stands for ‘medical records’, **NMT** stands for ‘neural machine translation’, **OT** stands for ‘online text’, **PA** stands for ‘private attributes’, **PET** stands for ‘privacy-enhancing technology’, **PHI** stands for ‘protected health information’, ***T*** stands for ‘target’, ***S*** stands for ‘scenario’, ***R*** stands for a set of records, **RE** stands for ‘regular expressions’, **RL** stands for ‘representation learning’, **SA** stands for ‘standard anonymization’, **SM** stands for ‘social media’, **S-PDTP** stands for ‘Sequential-PDTP’, **ST** stands for ‘style transfer’, **TL** stands for ‘transfer learning’, **VAE** stands for ‘variational autoencoder’, **VST** stands for ‘vector space transformation’, ***W*** stands for a set of target words.

information, such as names, dates, and locations, that may lead to the re-identification of a document collection, followed by the replacement of this information with artificial codes (e.g., ‘xxxx’) (Sánchez et al. 2018; Narayanan and Shmatikov 2008; Eder et al. 2020). Since text data is a rich source of private attributes, such as gender, ethnicity, location, and political views, text anonymization is a well-known challenge in the literature on privacy-preserving NLP. Mosallanezhad et al. (2019) propose an anonymizator based on reinforcement learning that extracts a latent representation of text and manipulates this representation to mask all private information it may hold. Simultaneously, the utility of the representation is preserved by changing the reinforcement learning agent’s loss function and assessing the quality of the embedded representation. Sánchez et al. (2018) describe a system for anonymizing images of printed documents whose text encompasses private information, such as names, addresses, dates, and financial content. To do so, the authors train a CNN model to strip private information out of images of invoices written in the Spanish language. Pablos et al. (2020) also conducted experiments for the anonymization of documents in the Spanish language but focused on clinical records, which were anonymized by variations of BERT. One of the biggest advantages of this language model is the prospect of outstanding performances without demanding feature engineering for the specific task. Therefore, the authors use this model as the basis for a sequence labeling approach that detects if each token in a sentence is related to a private feature or not. Finally, the results also demonstrate that BERT is robust to drops in the size of the training data, solely resulting in small performance reductions.

4.2.3 Differential privacy (DP) methods

Data instances can be individually protected by DP (Dwork 2008; Fernandes et al. 2019), which grants theoretical bounds to the protection of personal information in each data instance within a database, even though the aggregated statistical information of the whole database is revealed (Melamud and Shivade 2019). DP takes into account the assumption of plausible deniability (Fernandes et al. 2019), in which the output of a query may arise from a database that does not contain personal information as possible as from one that does. Ideally, there should be no way to distinguish between these two possibilities (Fernandes et al. 2019). For instance, models trained on documents featuring personal information will provide stronger DP guarantees the less their outputs rely on individual documents in the collection (Melamud and Shivade 2019). Formally, a randomized function $\hat{k}$ gives ϵ-DP in case for two collections C and C', which differ by at most one element, and all $\mathcal{S} \subseteq Range(\hat{k})$:

$$Pr[\hat{k}(C) \in \mathcal{S}] \leq exp(\epsilon) \times Pr[\hat{k}(C') \in \mathcal{S}]. \tag{2}$$

Every mechanism that satisfies this definition, which is depicted by Figure 8, will address worries about leakages of personal information from any individual element since its inclusion or removal would not turn the output significantly more or less likely (Dwork 2008). On the one hand, the efficiency of DP at protecting personal information often comes along with overheads in complexity and running time (Melamud and Shivade 2019). On the other hand, this mechanism presents noticeable flexibility in finding a balance between performance degradation and privacy budget.

DP provides theoretical bounds that preserve privacy, whereas regular data sanitization approaches may fail to demonstrate that privacy issues are formally averted. Further advantages of DP regard its ability to be blended with either utility task or NLP model. Feyisetan et al. (2020) perturb text data using a $d_{\mathcal{X}}$-privacy method, which is similar to a local DP setting when it comes to perturbing each data record independently. The $d_{\mathcal{X}}$-privacy grants privacy bounds onto location data, generalizing DP across distance metrics, such as Hamming distance, Euclidean, Manhattan, and Chebyshev metrics (Feyisetan et al. 2019). Then, calibrated noise is added to word representations from the word embedding models GloVe and fasttext. The method takes a string s of length $|s|$ as input and outputs a perturbed string s' of same length, which is privatized by a $d_{\mathcal{X}}$-privacy mechanism $\mathcal{M} : \mathcal{X} \rightarrow \mathcal{X}$, where $\mathcal{X} = \mathcal{D}^l$ represents the space of all strings with length l whose words are in a dictionary $\mathcal{D}$. Then $\mathcal{M}$ computes the embedding $\phi(w)$ for each word $w \in s$, adding calibrated random noise $\mathcal{N}$ to yield a perturbed embedding $\phi' = \phi(w) + \mathcal{N}$. Later on, w is replaced with a word w' whose embedding is the closest to ϕ', according to a Euclidean metric d. The authors consider that a randomized algorithm satisfies DP if its output distribution is similar to those when the algorithm is applied to two adjacent databases. They argue that the notion of similarity is managed by a parameter ϵ, which governs the extent privacy is preserved from full privacy, when it assumes the value of 0, to null privacy when it approaches ∞. For instance, $\mathcal{N}$ is sampled from a distribution z with density $\mathcal{P}_{\mathcal{N}}(z) \propto exp(-\epsilon||z||)$. By varying the values of ϵ, the tradeoff between privacy and utility is demonstrated.

Fernandes et al. (2019) obfuscate the writing style of texts without losing content, addressing the threats of unintended authorship identification. The authors assume that an author’s attributes can be predicted from the writing style, such as identity, age, gender, and mother tongue. A DP mechanism inspired by $d_{\mathcal{X}}$-privacy then perturbs bag-of-words representations of texts, preserving topic classification but disturbing clues that lead to authorship information. Firstly, a randomized function $\hat{k}$ receives b, b' bag-of-words as inputs and outputs noisy bag-of-words $\hat{k}(b), \hat{k}(b')$. If b and b' are classified as similar in topic, their perturbed versions $\hat{k}(b), \hat{k}(b')$ should also be similar to each other, depending on the privacy budget ϵ, regardless of authorship. Finally, $\hat{k}(b)$ should be distributed in agreement with a Laplace

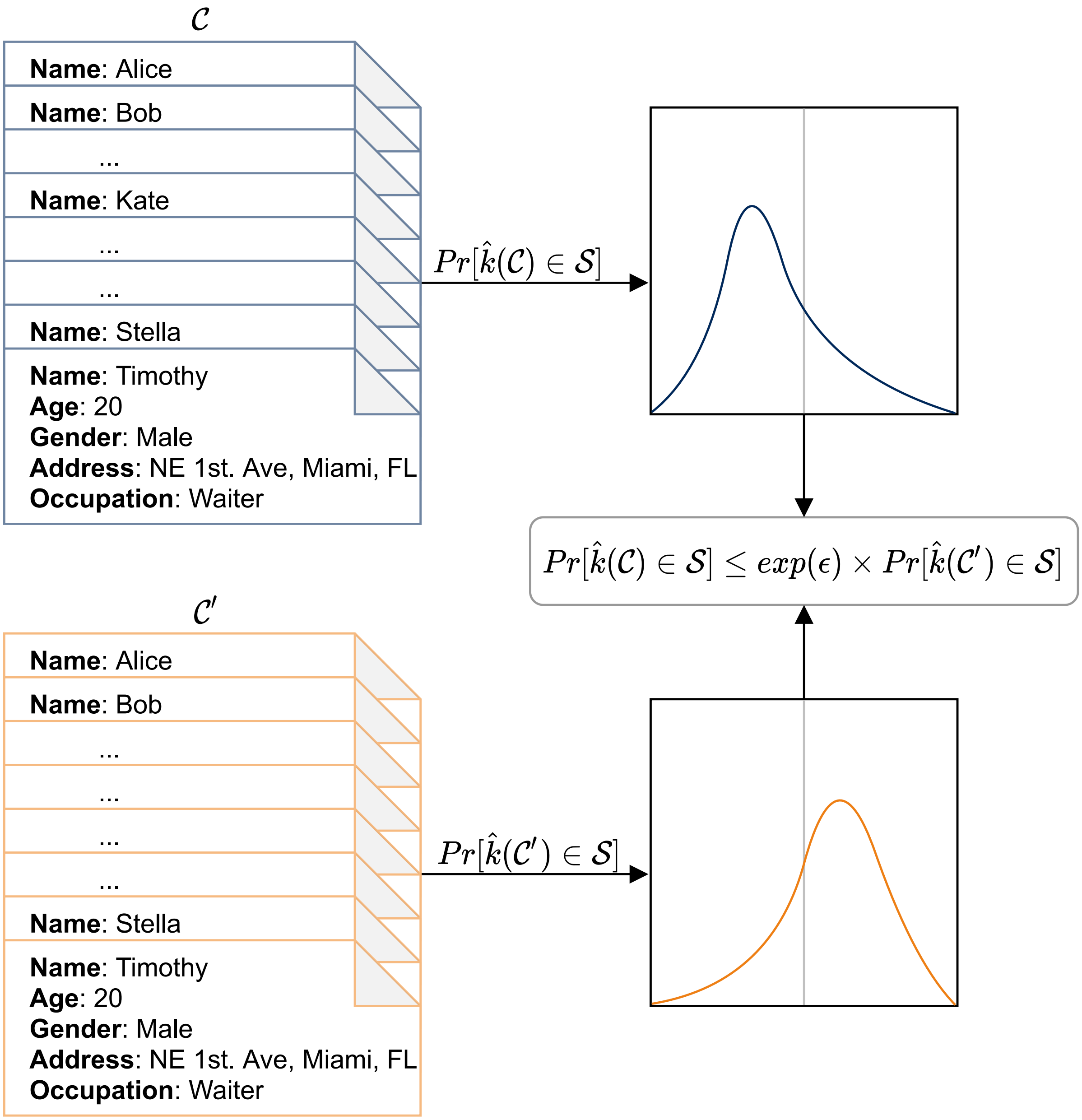

$\mathcal{C}$
Name: Alice
Name: Bob
...
Name: Kate
...
...
Name: Stella
Name: Timothy
Age: 20
Gender: Male
Address: NE 1st. Ave, Miami, FL
Occupation: Waiter
$Pr[\hat{k}(\mathcal{C}) \in \mathcal{S}]$
$Pr[\hat{k}(\mathcal{C}) \in \mathcal{S}] \leq exp(\epsilon) \times Pr[\hat{k}(\mathcal{C}') \in \mathcal{S}]$
$\mathcal{C}'$
Name: Alice
Name: Bob
...
...
...
...
Name: Stella
Name: Timothy
Age: 20
Gender: Male
Address: NE 1st. Ave, Miami, FL
Occupation: Waiter
$Pr[\hat{k}(\mathcal{C}') \in \mathcal{S}]$

Figure 8 Example of ϵ-DP for two collections C and C'. The collections differ by at most one element (i.e., the document for Kate).

probability density function calculated according to a metric for semantic similarities, such as the Earth Mover's distance.

Clinical notes are unstructured text data that encompass information input by doctors, nurses, or other patient care staff members. This kind of data requires de-identification before sharing activities, so patient privacy is not put at risk. Melamud and Shivade (2019) propose a method relying on DP to generate synthetic clinical notes, which can be safely shared. The authors define a setup for the task in three steps. Firstly, real de-identified datasets of clinical notes are used to train neural models that output synthetic notes. Secondly, privacy measures assess the privacy safeguarding properties of the synthetic notes. Finally, the utility of the generated notes is estimated using benchmarks.

DP also helps produce fair text representations as to demographic attributes. Lyu et al. (2020) provide a framework for learning deferentially private representations of texts, which masks private content words, whereas guarantees fairness by reducing discrimination towards age, gender, and five ‘person’ entities. The framework assumes data exchange between client and server parties and takes into account the threat of an eavesdropping attack which discloses private information from text representations yielded by a feature extractor from the client’s side and sent to a classifier on the server’s side. In order to protect the text representation from the eavesdropper, the training algorithm adds noise to the representations generated by the feature extractor. The same level of noise is added for both training and test phases, escalating the model robustness to noisy representations. Later on, the feature extractor f'' is also given to the client. Another mechanism proposed by the authors for their framework consists of a word dropout which masks words before the DP noise injection. Let x_i be a sensitive input composed of g words, and $\vec{\boldsymbol{I}}$ a dropout vector $\vec{\boldsymbol{I}} \in \{0, 1\}^g$. Therefore, dropout will be a word-wise multiplication of x_i with $\vec{\boldsymbol{I}}$. The number of zeroes in $\vec{\boldsymbol{I}}$ is defined by the dropout rate μ as $g \cdot \mu$. Additionally, combining word dropout with the ϵ-differentially private mechanism is useful to lower the privacy budget without drastically degrading the inference performance.

4.2.4 Representation learning methods

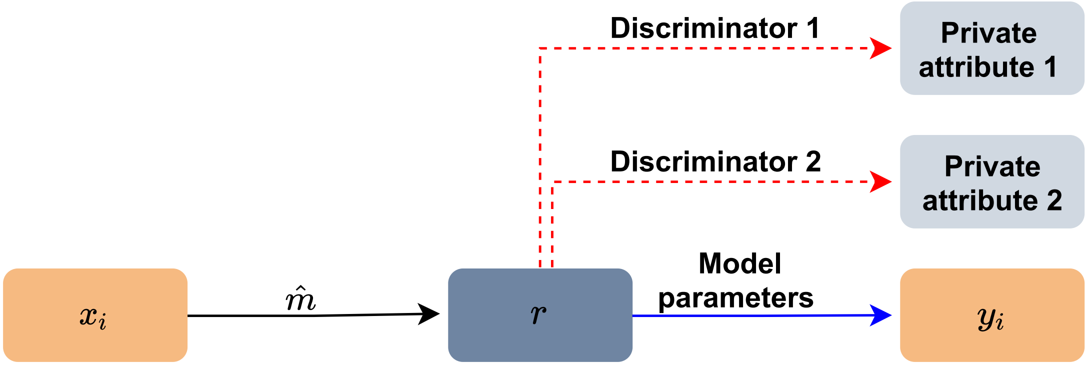


Figure 9 Adversarial learning for representation learning (Li et al. 2018). The method takes an input instance x_i and outputs a label y_i from the hidden representation r. The dashed red and blue lines represent the adversarial and standard losses.

Representation learning allows data to be represented as an information format conveniently used by DL models, whereas the efforts for manual feature engineering are no longer required (Bengio et al. 2013). Even text representations may still enclose private information and, for this reason, be prone to privacy threats. To address privacy issues, Li et al. (2018) train deep neural models using adversarial learning to yield unbiased representations and safeguard individuals’ private information, namely age, gender, and location. The proposed method takes inputs x to compute a hidden representation r, which is used to form the parametrization of a model $\hat{m}$ for predicting a target y, as depicted by Figure 9. During the training process, a loss function like cross-entropy is minimized to determine the model parameters θ_m. AL is based on learning a discriminator model $\hat{q}$ and $\hat{m}$ jointly. The discriminator model tries to predict a private attribute from each instance of r, so that the adversarial training can be seen as joint estimating the parameters θ_m and θ_q for $\hat{m}$ and $\hat{q}$ respectively. In order to safeguard privacy, r must lead to efficient predictions of y and deficient representations of the private attributes. Therefore, the objective function $\hat{\theta}$ of the method has the form

$$\hat{\theta} = \min_{\theta_m} \max_{\{\theta_{q_i}\}_{i=1}^{N}} \mathcal{E}(\hat{y}(x, \theta_m), y) - \sum_{i=1}^{N} (\lambda_i \cdot \mathcal{E}(\hat{a}(x, \theta_{c_i}), a_i)), \tag{3}$$

in which $\mathcal{E}$ represents the cross-entropy function, $\hat{y}$ denotes the predicted labels, a denotes a private attribute, $\hat{a}$ denotes the prediction of the discriminator model, and N is the number of private attributes. In the end, the learned text representations can be transferred for downstream applications as well as the discriminator.

Feyisetan et al. (2019) provide privacy preservation guarantees by perturbing word representations in Hyperbolic space, satisfying $d_{\mathcal{X}}$-privacy. The proposed method first transforms target words into vector representations. Second, these vectors are perturbed with noise sampled from the same Hyperbolic space where the vectors lie. The extent of this noise added to the vectors is proportional to the guarantees of privacy preservation. Finally, a post-processing step maps the vectors perturbed by noise into their closest words in the embedding vocabulary, hence preserving the semantic information. For instance, a query like 'how is the weather like in Los Angeles right now?' would have the location name replaced with the term 'city', based on similarity. Therefore, both user's query intent and privacy are preserved without semantic losses in the sense of the query.

Model interpretability can also be enhanced by representation learning at the pace of safeguarding private attributes. John et al. (2019) address the problem of disentangling the latent feature space of a neural model for text generation. The authors follow the premise that neural networks yield latent representations for the original feature set, which are not interpretable and do not present their meaning explicitly. So they came up with a method that is based on an autoencoder for encoding sentences into a latent space representation and learning to reconstruct these sentences to their original form. The representations produced by this method are then disentangled, i.e., divided into two parts with regard to different features (style and content) by a combination of adversarial and multi-task objectives. Sentiment associated with a sentence is considered as its style. Finally, the authors designed adversary losses to force the separation between the latent spaces for both style and content features, which can be used for text style transfer tasks later on.

4.2.5 Debiasing methods

Recent studies have brought evidence that pre-trained embedding models, ranging from word2vec to BERT, exhibit human biases towards demographic attributes, such as gender, race, nationality, and location. These biases play an influential role in downstream applications, which will be prone to making unfair decisions. NLP datasets may also encompass biases that weaken model generalization abilities when applied to unbiased datasets for transfer learning. For the sake of fairness in algorithmic decisions, debiasing consists of removing or lessening human biases that have the potential to compromise model decisions in NLP tasks. Among the types of bias, the one towards gender is broadly studied, mostly due to the popularity of word embedding models and the need for gender equality in systems relying on embeddings, which influence the everyday life of a huge number of people.

Bolukbasi et al. (2016) show that word2vec embeddings trained on the Google News dataset noticeably feature gender stereotypes like the association between the words 'receptionist' and 'female'. In order to get rid of such discriminatory associations while preserving the embedding power of solving analogy tasks and clustering similar concepts, the authors come up with an approach consisting of two phases. Firstly, evaluating whether gender stereotypes are present in vectors for occupation words, using crowd workers in the validation process, and generating analogy tasks where a word pair like {he, she} is used to predict new pair of words whose first term should be related to 'he', and the second one should be related to 'she'. So the results of this task are validated by the human workers to check if those analogies make sense and express gender stereotypes. A gender subspace is captured by the top component of a principal component analysis computed on ten gender pairs of difference vectors. Later on, the authors identify words that should be neutral with regard to gender, taking a set of 327 occupation terms. Finally, two debiasing algorithms are developed to two options: Soften or Neutralize and Equalize gender bias. Neutralize secures the gender-neutral words to be zero in the gender subspace, whereas Equalize balances the word sets outside the subspace and keeps the gender-neutral ones equidistant to both equality sets. For instance, given the equality sets {grandmother, grandfather} and {guy, gal}, introduced by the authors, the representation for the term 'babysit' should be equidistant to the terms in both equality sets after equalization. However, this representation should also be closer to those in the first equality set for the purpose of preserving the semantic relatedness of the terms.

Similarly, Kaneko and Bollegala (2019) propose a debiasing method for pre-trained word embeddings that is able to differentiate between non-discriminatory gender information and discriminatory gender bias. The authors argue that associations between words like 'bikini' and feminine nouns, or 'beard' and masculine nouns, would be expected and, then, capable of enhancing applications like recommender systems without prompting unfair model outcomes. On the other hand, profession titles such as 'doctor', 'developer', 'plumber', and 'professor' have frequently been stereotypically male-biased, but 'nurse', 'homemaker', 'babysitter', and 'secretary' have been stereotypically female-based. Therefore, they consider four information types, namely, feminine, masculine, gender-neutral, and stereotypical, to get rid of biases from stereotypical words, whereas gender information in feminine and masculine words and neutrality in gender neutral-words are maintained. Given a feminine regressor $\hat{u} : \mathbb{R}^e \to [0, 1]$, which has θ_u parameters for predicting the extent of femininity the word w presents. In this sense, highly feminine words are assigned to femininity values nearing 1. In a similar manner, a masculine regressor $\hat{v} : \mathbb{R}^e \to [0, 1]$ with parameters θ_v estimates the masculinity degree of w. Therefore, the debiasing function will be learned as the encoder component of an autoencoder $E : \mathbb{R}^d \to \mathbb{R}^e$ with parameters θ_E, whereas the decoder component is defined as $\mathcal{Z} : \mathbb{R}^d \to \mathbb{R}^e$ with parameters $\theta_{\mathcal{Z}}$. The number of dimensions of the original vector space is denoted by d, whereas the number of dimensions of the debiased vector space is given by e.

Word embedding models can pass gender biases on from training corpora to downstream applications. Font and Costa-jussà (2019) come up with a method to equalize gender bias in the task of neural machine translation using these word representations. The authors detect biases toward terms originally in English, which are translated into masculine forms in Spanish. For instance, the word 'friend' would be translated into a masculine Spanish word if it came along in a sentence with the term 'doctor', whereas it would be translated into the feminine form in case it was used in the same sentence as the term 'nurse'. The authors use a state-of-the-art transformer model for neural machine translation input with word embedding yielded by three embedding models, namely, GloVe, GN-GloVe, and Hard-Debiased GloVe. Additionally, they compare two different scenarios, featuring no pre-trained embeddings or using pre-trained embeddings from the same corpus on which the model is trained. The transformer models for the second scenario have three distinct cases regarding the use of pre-trained embeddings: only on the model encoder's side, only on the model decoder's side, or on both model sides. The experimental results show that debiased embedding models do not slash the translation performance.

Further applications for debiasing approaches include the detection of biases in text data with content related to politics and bullying online. Papakyriakopoulos et al. (2020) develop a method for bias detection in the German language and compare bias in embeddings from Wikipedia and political-social data, proving that biases are diffused into machine learning models. The authors test two methodologies to debias word embedding yielded by GloVe, and employ biased word representations to detect biases in new data samples. Gencoglu (2020) proposes a debiasing model for cyberbullying detection on different online media platforms, employing fairness constraints in the training step. The author conducts experiments on gender bias, language bias, date bias (e.g., drop in performance on recently created insult terms), and bias towards religion, race, and nationality. A sentence-DistilBERT is used to extract representations for posts and comments in the datasets. The objective function of the neural model was adapted to implement fairness measures.

4.2.6 Adversarial learning methods

Health data, like patient notes, is a widely known source of protected attributes to be taken into obliviousness by adversarial learning. Automatic de-identification approaches for PHI data are costly since massive datasets for model training are barely available due to regulations that hinder the sharing of medical records. Friedrich et al. (2019) present a method to yield shareable representations of medical text, without putting privacy at risk, by PHI removal that does not demand manual pseudonymization efforts. Firstly, adversarial learning-based word representations are learned from publicly available datasets and shared among medical institutions afterward. Secondly, the medical institutions convert their PHI raw data into these representations (e.g., a vector space) that will be pulled into a new dataset for de-identification, avoiding the leakage of any protected attributes. Finally, the approach is argued to provide defenses against plain-text and model inversion attacks.

Recent approaches of adversarial learning to safeguard training data include the removal of demographic attributes, such as gender, ethnicity, age, location, nationality, social status, and education level. Language models that encode these attributes are prone to a series of privacy issues that compromise their safety and fairness. Elazar and Goldberg (2018) demonstrate that demographic information can be encoded by neural network-based classifiers. Later on, an attacker network that predicts protected attributes above chance level is used to retrieve the demographic ones from the latent representation of the classification models. The authors use an adversarial component in order to pull out the attributes of ethnicity (race), gender, and age from tweets collected for the tasks of binary tweet-mention prediction and binary emoji-based sentiment prediction. In their configuration, a classifier was trained to predict the protected attributes alongside a one-layer LSTM meant to encode a sequence of tokens and undermine the classifier, namely an MLP. Therefore, the learned representations have their information with regards to the tasks maximized, while it is minimized to the protected attributes. As a result, the adversarial learning method demonstrates efficiency in avoiding leakages but fails to completely remove protected attributes from the text.

Barrett et al. (2019) revisit the experiments of Elazar and Goldberg (2018), analyzing correlations between the yielded representations on the models and the demographic attributes of age and gender. They introduce three correlation types. First, prevalent correlation arises from features associated with gender in most contexts, like in the sentence fragments including expressions like 'as a mother', 'my girlfriend', 'as a guy', 'the mailman', etc. Second, sample-specific correlation is related to features tied up with different demographic attributes depending on the domain or sample, such as the word 'bling' related to different ranges of ages if it is used to describe jewelry items, movies, or rap songs. Finally, accidental correlation demonstrates the relationship between text features and protected attributes in a particular dataset, although their uncommon relation. The experimental evaluation suggests that the model relies on spurious or accidental correlations limited to a specific sample of data since they fail on new data samples or domains.

Xu et al. (2019) come up with a privacy-aware text rewriting method for obfuscating sensitive information prior to data release to promote fair decisions which do not take into account demographic attributes. The authors defined this task as protecting the sensitive information of data providers by text rephrasing, which lessens the leakage of protected attributes, maintains the semantics of the original text, and preserves the grammatical fluency. Formally, this task assumes a set of inputs $X = \{x_1, x_2, \ldots, x_n\}$, in which each input x_i represents a word sequence $\langle w_1, w_2, \ldots, w_n \rangle$ associated with a sensitive attribute $a \in A$. It aims at finding a function $\hat{f}(x_i) : x_i \rightarrow y_i$, which translates x_i into a different word sequence $y_i \in Y$ that halts an attacker from detecting the values of a given the translated text. Since there is no parallel corpus to recognize patterns of privacy-preserving text rewriting, the authors approached the task as a monolingual machine translation problem, using back-translation. Here, a text is translated from English to French and later back to English. This task aims at minimizing the reconstruction loss between $\hat{f}(x_i)$ and y_i along with the risk loss towards privacy $R(X, Y, A)$. Two different obfuscation methods are proposed: one based on adversarial learning and the other based on fairness risk measurement. Adversarial learning is employed to yield representations that ease reconstructing the input texts while slashing the prediction of sensitive attributes by a linear classifier, which receives the yielded latent representations of the word sequences as inputs. In other words, the text reconstruction performance is maximized, whereas that of the linear classifier is minimized. On the other hand, fairness risk measurement concerns the discrepancy between the privacy-preserving translator and a subgroup translator that relies on a sensitive group attribute a. The lower the discrepancy, the better the obfuscation. A transformer (Vaswani et al. 2017) architecture is used for translation in the experimental evaluation, which aims at confounding the attributes of gender, race, and political leaning. Additionally, the evaluation of the leakage risk is estimated by logistic regression with L2 regularization (Pedregosa et al. 2011). Finally, the authors propose metrics for privacy-aware text rewriting to assure the requirements of fluency, obfuscation of sensitive information, and semantic relevance. Of the two proposed methods for the task, the one based on fairness risk preserves fluency and relevance to a greater degree than the adversarial one.

Other NLP tasks, such as sentiment analysis and topic classification, also pose privacy risks regarding adversarial attacks that have the potential to recover sensitive information from language representations. Coavoux et al. (2018) study this kind of privacy attack, propose privacy measures that gauge the leakage risk of private attributes from hidden representations learned by neural networks, discuss the privacy-utility tradeoff, and propose safeguarding methods by adding terms to the objective functions of the models. Both tasks of sentiment analysis and topic classification are approached in the experiments by an LSTM model. Moreover, Kumar et al. (2019) bring evidence that language classification models can learn topical features which are confounds for an inference task of native language identification. Hence, the authors propose an adversarial learning setting for representing the latent confounds. At the same time, a BiLSTM model with attention obfuscates these features by predicting them along with the actual labels for each input. This method is argued to be less prone to using a smaller amount of information related to confounds, besides better generalization abilities, and enhanced for learning writing style features instead of content ones.

Sweeney and Najafian (2020) use adversarial learning to remove correlations between demographic attributes and sentiments in word vectors, referring to this problem as sentiment bias. For the authors, word vectors for demographic identity terms, such as those related to nationality, religion, gender, and names, should retain neutrality with regard to sentiments. In the experiments, they, firstly, retrieve the vectors for positive words from the Sentiment Lexicon dataset (Hu and Liu 2004) to create a matrix for taking the most significant component from the principal component analysis. The same process is done for the negative words in the dataset afterward since there is no pre-defined pairwise mapping between positive and negative terms in a similar fashion as the data for tackling gender bias. Another reason for this two-step process relates to the semantics for drawing differences between positive and negative perceptions, which is considerably looser than that for gender. Secondly, the signed difference between both negative and positive components is taken and named the directional sentiment vector. Further, the sentiment polarity of all remaining vectors is assessed by projecting these against the directional sentiment ones. In the following stage, a classification model is used to check if the directional sentiment vectors are able to hold sentiment polarity. Adversarial learning computes two distinct objectives. The first finds the least square distance between the input word vector and its debiased version. Simultaneously, the adversarial one predicts the sentiment polarity based on the input vector. Finally, the embedding models word2vec and GloVe are debiased and later tested for the downstream tasks of sentiment valence (intensity) regression and toxicity classification, leading to semantic preservation and fair decisions.

4.3 Trusted methods

When DL models are designed for learning over data or untrusted computation scenarios, trusted methods appear as solutions. We have identified trusted methods for scenarios involving cloud environments and distributed parties, as depicted by Figure 10 and summarized in Table 7. The sub-categories and groups of trusted methods are populated with well-known PETs, often implemented in real-time systems, such as federated models, transfer learning, and encryption.

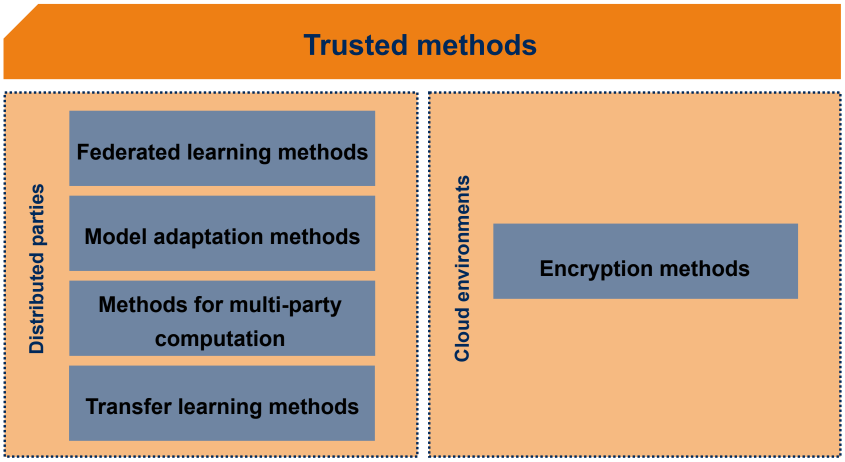


Figure 10 Sub-categories and groups for trusted methods

4.3.1 Federated learning (FL) methods

FL is a technique proposed by Google in 2016 (Konečný et al. 2016; McMahan et al. 2017) which enables centralized models to train on data distributed over a huge number clients (Konečný et al. 2016). For instance, a model for spell-checking in virtual keyboards can be hosted on a central server and trained in a decentralized manner across smartphones, without any data exchanges between the client devices and the server or between client devices. FL does not allow data to ever leave its owner's device (Chen et al. 2019). This learning paradigm consists of training rounds where every client updates the model it receives from the central server with computations on its local data and passes this update on to the server, which computes an improved global model upon the aggregation of all client-side updates (Konečný et al. 2016). Figure 11 illustrates an example of an FL setting for mobile devices.

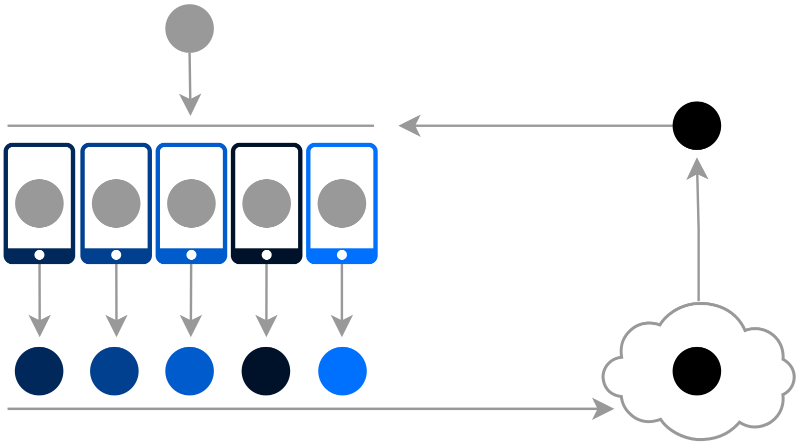


Figure 11 Federated learning setting for mobile devices (McMahan and Ramage 2017). From the initial global model at the top, each mobile device locally computes its model update and sends it to the server (lower right corner), which aggregates all the received local updates and distributes the updated global model to the mobile devices.

Moreover, FL is hoped for text datasets that encompass user-generated text, financial transactions, medical records, personal preferences, trajectories, and so on (Zhu et al. 2020). Formally, FL learns a model whose parameters are held by a matrix $\boldsymbol{M}$ from data samples stored by many different clients, sharing the current model $\boldsymbol{M}_{t'}$ with a set T of n' clients, which updates the model with their local data at each training round $t' \geq 0$ (Konečný et al. 2016). Each selected client T_i sends its update $\boldsymbol{H}_{t'}^{T_i} := \boldsymbol{M}_{t'}^{T_i} - \boldsymbol{M}_{t'}$ back to the server, which aggregates all the client-side updates

Table 7 Summary of trusted methods

Group	**Work**	**Neural models**	**PET**	***T***	***S***
FL methods	Chen et al. (2019)	CIFG LSTM, GLSTM	FL	*W*	VK
	McMahan et al. (2018)	LSTM	(ϵ, δ)-DP, FL	*W*	VK
	Hard et al. (2018)	CIFG LSTM	FL	*W*	VK
	Huang et al. (2020)	BERT, MLP, RoBERTa	TextHide	*W*	DC
	Zhu et al. (2020)	TextCNN	(ϵ, δ)-DP, FL	*W*	DC
	Qi et al. (2020)	GloVe, CNN, GRU	(ϵ, δ)-DP, FL	*W*	NR
MA methods	Clinchant et al. (2016)	MD autoencoders	Transductive adaptation	*D*	PI
	Zhao et al. (2018)	GN-GloVe	Private training	*W*	PA
Methods for MPC	Feng et al. (2020)	Seq2seq + attention	MPC protocols	*W*	MSD
TL methods	Alawad et al. (2020)	MT-CNN	Vocabulary restriction	*W*	CT
	Martinelli et al. (2020)	Fasttext	SID	*D*	PI
	Hu and Yang (2020)	PrivNet	Adapted loss	*D*	PI
Encryption Methods	Dai et al. (2019)	Doc2Vec	Searchable encryption	*D*	Cloud
	Liu and Wang (2020)	LSTM	Searchable encryption	*K*	Cloud

CT stands for 'collaborative training', ***D*** stands for a set of documents, **DC** stands for 'document collection', **DP** stands for 'differential privacy', **FL** stands for 'federated learning', **GRU** stands for 'Gated recurrent units', ***K*** stands for a set of keywords, **MA** stands for 'model adaptation', **MD** stands for 'marginalized denoising', **MPC** stands for 'multi-party computation', **MSD** stands for 'multi-source data', **NR** stands for 'news recommendation', **PA** stands for 'private attributes', **PET** stands for 'privacy-enhancing technology', **PI** stands for 'private inference', ***S*** stands for 'scenario', **SID** stands for 'sensitive information detection' ***T*** stands for 'target', **TL** stands for 'transfer learning', **VK** stands for 'virtual keyboards', ***W*** stands for a set of target words.

and comes up with the global update in the form:

$$\begin{aligned} \boldsymbol{M}_{t'+1} &= \boldsymbol{M}_{t'} + \eta_{t'}\boldsymbol{H}_{t'}, \\ \boldsymbol{H}_{t'} &:= \frac{1}{n'}\sum_{T_i \in T} \boldsymbol{H}_{t'}^{T_i} \end{aligned} \tag{4}$$

in which $\boldsymbol{M}_{t'}^{T_1}, \boldsymbol{M}_{t'}^{T_2}, \ldots, \boldsymbol{M}_{t'}^{T_{n'}}$ are the updated local models, and $\eta_{t'}$ is the learning rate (Konečný et al. 2016). FL is especially suitable for scenarios in which the client devices may not feature a high-speed bandwidth, such as smartphones or internet of things sensors, but reducing the communication cost, whereas preserving data privacy is still an open challenge for federated models.

FL methods became popular for smartphone applications, like virtual keyboards. Chen et al. (2019) propose a federated training for an RNN using the FederatedAveraging (McMahan et al. 2017) algorithm and approximate this server-side model with an n-gram language model, which allows faster inference on the client's side. For reasons concerning memory and latency, language models for virtual keyboards are based on n-grams and do not surpass ten megabytes in size. Given previously typed words $w_1, w_2, \ldots, w_{n-1}$, the language model will assign a probability to predict the next word as

$$Pr(w_n \mid w_{n-1}, \ldots, w_1). \tag{5}$$

The authors follow the assumption that n-gram language models are Markovian distributions of order $o - 1$, in which o represents the order of the n-gram, in the form

$$Pr(w_n \mid w_{n-1}, \ldots, w_{n-o+1}). \tag{6}$$

FederatedAveraging collects unigrams on each client device, returning counting statistics to the server instead of gradients. A unigram distribution is counted based on a white-list vocabulary. Later on, the unigram part of an n-gram is replaced with its distribution, producing the final language model. Then, a modified SampleApprox. algorithm (Suresh et al. 2019) approximates the RNN. Tests are conducted on text data from virtual keyboards for two languages, namely American English and Brazilian Portuguese. Finally, the results demonstrate that federated n-gram models present high quality for faster inference than server-based models, with the advantage of keeping private user-generated data on their owner's smartphone.

Additional methods for next word prediction in virtual keyboards include McMahan et al. (2018), which apply DP alongside FL training. The authors present a version of the FederatedAveraging algorithm that is perturbed by noise, satisfying user-adjacent DP (Abadi et al. 2016). This approach presents strong privacy guarantees without losses of utility since it does not decrease the performance of the target task drastically. Hard et al. (2018) also explore this task, comparing a server-based training using stochastic gradient descent against client-side training that uses the FederatedAveraging algorithm. The results demonstrate the efficiency of the federated training alongside gains in prediction recall.

NLP tasks also feature privacy risks that should be tackled by NLP methods, such as eavesdropping attacks or the inversion of general-purpose language models like BERT. Huang et al. (2020) create the framework TextHide for addressing these challenges in natural language understanding by protecting the training data privacy at a minimum cost concerning both training time and utility performance. This framework requires each client in a federated training setting to add a simple encryption step to hide the BERT representations of its stored text. Therefore, an attacker would have to pay a huge computational cost to break the encryption scheme and recover the training samples from the model.

Recent challenges in FL include the protection of the parties that are involved in the decentralized model training for sentence intent classification (Li et al. 2008). Taking into account the numerous applications that rely on this task, such as review categorization and intelligent customer services, Zhu et al. (2020) show how to adapt the NLP model TextCNN (Kim 2014) for federated training, adding Gaussian noise to the model gradients before updating the model parameters. TextCNN is built upon a CNN for classification tasks at the sentence level. For FL model training, the central server receives the gradients, computed on the local data stored by the clients, at the end of each epoch. Firstly, given the values for the parameters on the central server's model, each client samples its data in a batch by batch manner. Secondly, the local parameters are updated based on the gradients computed for each sample batch. At the end of the iterations over all the sample batches, the cumulative difference of the parameter values is then sent to the central server for cross-client aggregation and updating the parameters of the global model. Before sending the locally computed gradients to the server, the privacy accountant is computed, and controlled noise is added to these gradients. This procedure protects the per-sample privacy of each client involved in the federated training. Therefore, each client controls its privacy budget instead of the central model and stops its updates to the server once the privacy threshold is reached. The authors argue that their method is convenient for scenarios in which the clients trust the communication channels, e.g., by encryption. However, the central server is an honest-but-curious one. Sensitive information, for this reason, should not be exposed to the server.

FL can also be blended with local DP for news recommendations without storing user data in a central server. Qi et al. (2020) approach this task, proposing a framework that first keeps a local copy of the news recommendation model on each user's device and computes gradients locally based on the behavior of the users in their devices. Second, the gradients of a randomly selected group of users are uploaded to the server for aggregation and subsequent updating of the global model that it stores. Prior to the gradients' upload, the framework applies local DP to perturb implicit private information they may hold. Finally, the updated global model is shared with the user devices, which will compute their updates locally.

4.3.2 Model adaptation methods

Data is often a target of legal provisions and technical constraints that require the adaptation of machine learning and DL methods to meet privacy preservation requirements that may lead to penalties in case of noncompliance. For instance, some domains present huge amounts of data alongside high costs to acquire labels to perform classification tasks. In such backdrops, domain adaptation (DA) methodologies can be employed but sometimes result in privacy issues. Clinchant et al. (2016) apply a marginalized denoising autoencoder in a transductive manner on text data that suffered from domain shift during DA. When the source and target domains differ, performance downsides on the latter domain can be noticed, especially if there is no known label information. Therefore, this autoencoder aims to minimize the following reconstruction loss:

$$\mathcal{L}(F) = \sum_{i=1}^{n} \sum_{c=1}^{C} ||x - F\tilde{x}_{ic}||^2, \tag{7}$$

where F is a linear mapping between the two domains, n is the number of inputs x, $\tilde{x}$ denotes an input that was corrupted C times by random dropout of features. The transductive adaptation mechanism proposed by the authors consists of leveraging the feature space for the target data using the class scores generated by the model trained on the source domain, exploiting correlations, and protecting the source data's content. Experiments were conducted on two DA datasets and showed the effectiveness of their adapted autoencoder over a standard classifier baseline.

Word embeddings are also prone to hazards from sensitive data, such as biases that play a role in discriminatory decisions made by application systems like resume filters. Hence adaptation requirements for these neural models are also approached in the literature. Zhao et al. (2018) adapt GloVe embeddings to cope with protected attributes during the training pace, yielding gender-neutral word representations without undermining their functionality. The proposed method safeguards sensitive attributes in specific dimensions that can be easily stripped away afterward. Masculine and feminine words are used as seeds for restricting gender information in the learned vector spaces, as demonstrated by a list of profession titles that are gender-neutral by definition. However, their original GloVe representations exhibited gender stereotypes. Cosine similarity is used to quantify the gender tendency of each word vector $\vec{\boldsymbol{w}}$ and the gender direction v' in the form

$$\frac{\vec{\boldsymbol{w}} \cdot v'}{||\vec{\boldsymbol{w}}||\ ||v'||}. \tag{8}$$

The gender direction variable v' averages the differences in the representations for feminine words and their masculine counterparts in a predefined set of word pairs Ω' as

$$v' = \frac{1}{||\Omega'||} \sum_{(w_v, w_u) \in \Omega'} (\vec{\boldsymbol{w}}_{\boldsymbol{v}} - \vec{\boldsymbol{w}}_{\boldsymbol{u}}), \tag{9}$$

where w_v and w_u are respectively a masculine and a feminine word for which the vector representations $\vec{\boldsymbol{w}}_{\boldsymbol{v}}$ and $\vec{\boldsymbol{w}}_{\boldsymbol{u}}$ were generated. A smaller similarity to gender direction suggests diminished gender information in the vector space. This method enables unbiased representations of sensitive words as to binary genders and has the potential to be extended to additional features such as sentiments and demographic information.

4.3.3 Methods for multi-party computation (MPC)

Private data often arises from multiple providers, such as distributed databases, users, companies, and devices in internet of things. MPC is a PET applicable to such scenarios defined as a generic cryptographic primitive for secure computations of agreed functions over different private data sources, which should not be revealed (Zhao et al. 2019; Feng et al. 2020). In contrast, the computation results hold an interest for all the parties and can be made available therein, assuring correctness and privacy properties (Cramer et al. 2015). For instance, when the exchange of plaintext

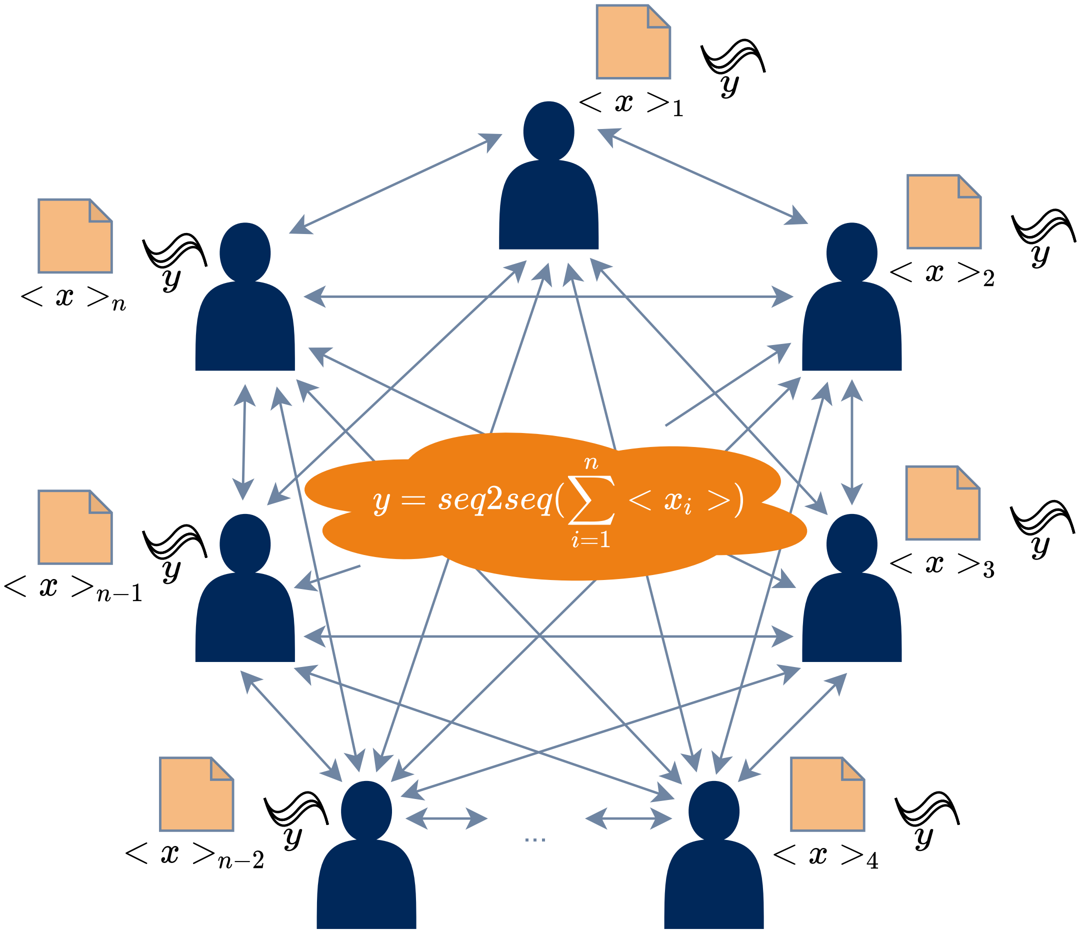


Figure 12 MPC framework designed by Feng et al. (2020). The framework involves multiple parties which can obtain the publicly available trained seq2sec model. Secret sharing is used to compute the NLP task. $< x >_i$ denotes the knowledge held by each party p_i, and y represents the model output.

is not allowed, MPC can be used to share keywords extracted from many text file sets to enrich NLP applications (Feng et al. 2020). Formally, there will be a set of inputs $\{x_1, x_2, \ldots, x_n\}$ so that each party p_i will hold x_i and agree to compute $y = \hat{f}(x_1, x_2, \ldots, x_n)$, in which y is the output information for release, and $\hat{f}$ is the agreed function on the whole set of inputs (Cramer et al. 2015). The inputs may include keywords, messages, and medical records.

Privacy in NLP models can also be achieved by MPC protocols that enable separate computations on secret inputs hailing from diverse parties, such as users, devices, and service providers. The parties involved in the computations should not learn from each other's inputs but their outputs instead since there should be no plaintext exchange. Feng et al. (2020) design new MPC protocols aiming to preserve the privacy of every computation party in the NLP task of neural machine translation, simultaneously computing non-linear functions for deep neural networks quicker. Figure 12 depicts the framework proposed by the authors. The authors come up with interactive MPC protocols, using both additive and multiplicative secret sharing, for the non-linear activation functions of sigmoid σ and tanh τ, which are defined for any input value x, respectively, as

$$\sigma(x) = \frac{e^x}{e^x - 1} \tag{10}$$

and

$$\tau(x) = \frac{e^{2x} - 1}{e^{2x} + 1}. \tag{11}$$

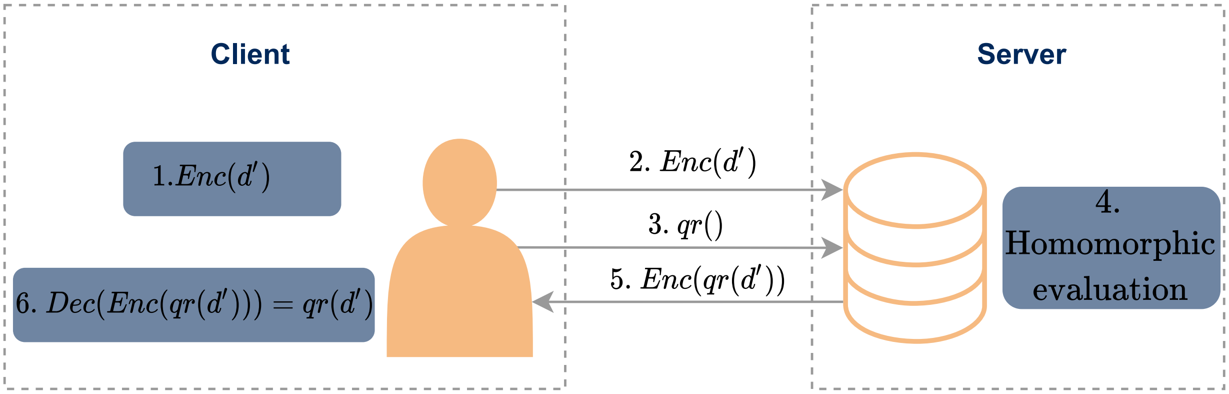


Figure 13 Simple homomorphic encryption scenario (Acar et al. 2018). In this scenario, d' denotes a document. **1.** The client encrypts d'. **2.** The client sends its encrypted document $Enc(d')$ to the server. **3.** The client sends a query function $qr()$ to the server for querying its data. **4.** The server performs a homomorphic operation over the data without decryption. **5.** The server sends the encrypted results to the client. **6.** The client recovers the data with its private key and retrieves the query function results $qr(d')$.

The protocols are implemented on an RNN-based seq2seq with an attention model, which performs predictions on multi-source data, keeping parties and attackers from learning the secret knowledge of another party during the model inference step.

4.3.4 Transfer learning methods

In domains that lack data for specific tasks and prevent data sharing, transfer learning (TL) is a straightforward solution relying on models that allow knowledge transfer by applying neural networks to tasks that differ from those targeted by previous training (Weiss et al. 2016). TL approaches can be adjusted for preserving private or sensitive information arising from medical text, documents, social media posts, etc. Taking the medical domain as an example, the vocabulary of NLP models may contain specific terms that breach the anonymity of documents used in the training step, so protective measures have to be set prior to the release of these models for further fine-tuning. Alawad et al. (2020) implement a multi-task CNN for TL on cancer registries aiming at information extraction from pathology reports concerning six characteristics (i.e., tumor site, subsite, literality, behavior, histology, and grade). Regarding privacy-preserving NLP, the authors come up with a restriction for the word embedding vocabulary to keep out PHI, such as patient names, therefore allowing the model to be shared between different registries observing data security. This vocabulary restriction consists of restraining the shareable tokens in the embedding vocabulary to those obtained from word embeddings pre-trained on corpora with no protected textual information like vectors trained on PubMed and MIMIC-III datasets (Zhang et al. 2019). Alongside the PET, the authors test two transfer learning settings: acyclic TL and cyclic TL. The former regards a usual TL approach in which a model is trained on a registry and forwarded to the next one for adjustments, whereas the latter embodies iterations between all the involved registries during the training step until the model converges.

Recognition of sensitive information is another use case suitable for TL approaches since each application field imposes its categories of sensitive content whose identification mostly relies on efforts spent by humans. Martinelli et al. (2020) propose a methodology to improve NLP models for knowledge transfer across general and specific domain tasks in steps prior to data obfuscation. From completely unlabeled corpora of documents, the authors are able to yield annotated versions whose sensitive content has been recognized and tagged. In the approach, word vectors extracted from fasttext make up the document representations to be combined with a sensitive content detection technique based on named-entity recognition and topics extraction. Therefore, the approach's outcomes may lighten the workloads of human officers that perform sensitive content detection manually.

4.3.5 Encryption methods

Encryption is a broadly used PET to encode information when sharing data in raw format is not a secure option. So a ciphertext is yielded by applying an encryption function over a data instance in the so-called plaintext. FHE has emerged as an encryption scheme that allows computations on encrypted data with no requirements of decryption (Gentry 2009). In other words, it allows mathematical operations to be computed on encrypted data while it is still in its encrypted form (Acar et al. 2018), as depicted by Figure 13. Consequently, outsourced models hosted on untrusted cloud environments are able to perform training or inference on encrypted datasets without disclosing any clues about their content. However, FHE frequently leads to computation overheads due to the amount of noise used added to the ciphertexts, although its efficiency for privacy protection (Li and Huang 2020). Another drawback of this encryption scheme is the need for approximating activation functions, such as relu, sigmoid, and tanh, since only addition and multiplication operations are supported.

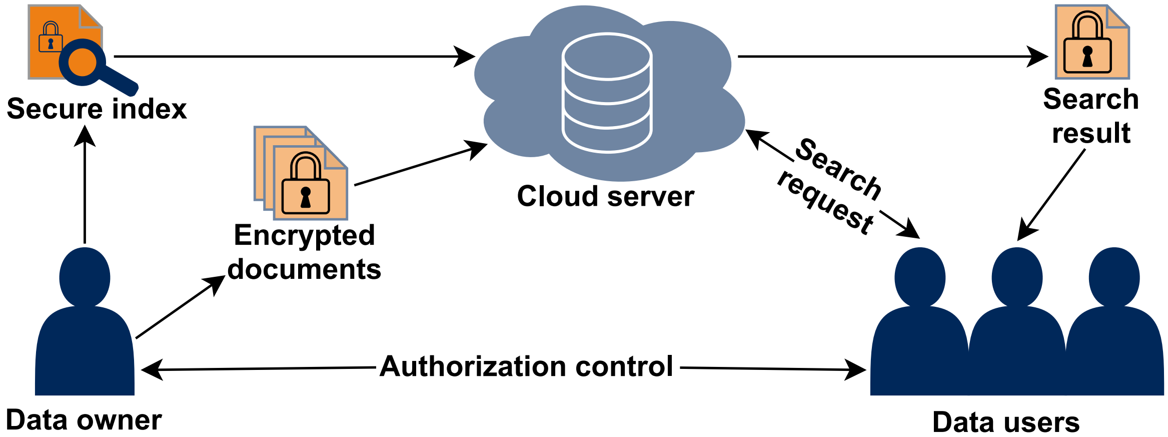


Figure 14 Searchable encryption system proposed by Dai et al. (2019). First, the data owner encrypts its documents and sends them to the cloud server in this system. Second, the data owner also encrypts document vectors and uses them as the secure index, which is also sent to the cloud server. Third, data users who have the authorization to query the encrypted documents encrypt their queries and send them to the cloud server. Further, the cloud server returns the encrypted document results to the data users. Afterward, the data users decrypt the document results using the secret key shared by the data owner. Finally, the query is finished.

Another encryption scheme that enables utility tasks is searchable encryption (SE), which encrypts document collections in such a manner that search capabilities can be delegated by the data owner without the need for decryption by the server or service provider (Cash et al. 2015). Hence, an untrusted server can provide searches for uses without disclosing the content of both data and queries (Liu and Wang 2020). Figure 14 shows an example of a SE system. Moreover, a common challenge on SE includes preserving the semantic relations between words and documents, which may undermine search results due to ambiguities in the keywords (Dai et al. 2019). Generally, SE involves four algorithms for, respectively, key generation, encryption, token generation, and search (Liu and Wang 2020). Nonetheless, this encryption scheme is sensitive to attacks that aim at recovering encrypted keywords.

Dai et al. (2019) propose two privacy-preserving keyword search schemes on data protected by SE, namely DMSRE and EDMRSE. Both schemes rely on the embedding model doc2vec (Le and Mikolov 2014) for document representation. DMSRE is composed of five procedures. Firstly, it generates a secure key. Secondly, it pre-trains the embedding model and extracts feature vectors for every document d' in the set of documents D. It also processes d' and extracted vectors at the same time that it obtains the encrypted documents $\tilde{\boldsymbol{D}}$ and the encrypted document indexes $\tilde{\boldsymbol{I}}$ for later searches. Subsequently, the trapdoor $\tilde{\boldsymbol{V_Q}}$ is produced for the queried keywords $\boldsymbol{Q}$ from a data user. Finally, it conducts the inner product between every document index in $\tilde{\boldsymbol{I}}$ and $\tilde{\boldsymbol{V_Q}}$ in order to retrieve the k most semantically related results as:

$$| \boldsymbol{RList} |= k \forall \tilde{d}_i, \tilde{d}_j (\tilde{d}_i \in \boldsymbol{RList} \wedge \tilde{d}_j \in (\tilde{\boldsymbol{D}} - \boldsymbol{RList})) \rightarrow \tilde{\boldsymbol{I}}_i \cdot \tilde{\boldsymbol{V_Q}} > \tilde{\boldsymbol{I}}_j \cdot \tilde{\boldsymbol{V_Q}}. \quad (12)$$

EDMRSE has its security enhanced by adding phantom terms on both document vectors and trapdoors to confound the search results and keep the cloud model from gathering statistical information about the documents. Except for the procedure of pre-training the embedding model and extracting the document's vector, all the remaining ones have different definitions to deal with the introduced phantom terms. As a result, EDMRSE increases privacy protection but decreases search accuracy.

Although providing efficient schemes for privacy protection of text data, SE is prone to file injection attacks in which a malicious adversary injects customized files into the encrypted search system and distinguishes specific keywords by observing the patterns on the encrypted versions of the injected files. Liu and Wang (2020) investigate the susceptibility of leakages from data protected by a SE scheme using an LSTM model to generate text files to be injected into the encryption scheme. The authors conducted extensive experiments finding that automatically generated texts present low quality that could be manually identified by humans and also diminish the feasibility likelihood of attacks that use them as injected files. Automatic file injection attack detection was performed by three ensemble methods, i.e., random forest, Adaboost based on support vector machines, and Adaboost based on random forest. Among those methods, the third one provided the highest accuracy rates in the study. The main takeaway from this work regards the practicability to identify automatically generated files for injection attacks, although semantically meaningful files injected in an ad-hoc manner cannot be easily detected and attain successful attacks.

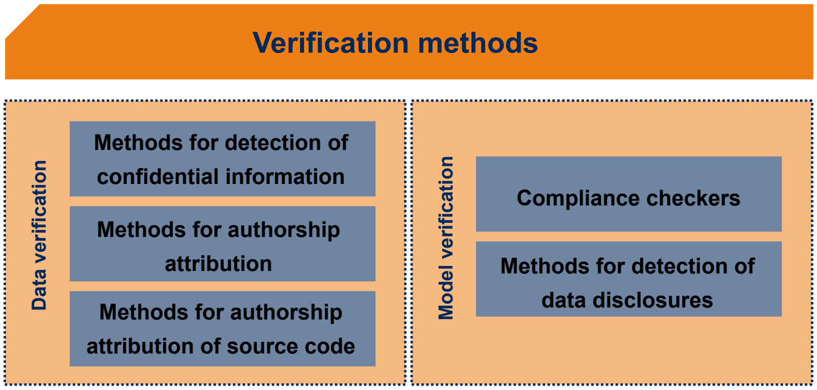

Verification methods
Data verification
Methods for detection of confidential information
Methods for authorship attribution
Methods for authorship attribution of source code
Model verification
Compliance checkers
Methods for detection of data disclosures

Figure 15 Sub-categories and groups for verification methods

4.4 Verification methods

Verification methods encompass approaches for verifying data and model susceptibility to privacy threats. These methods are mostly applied during post-processing evaluations. Figure 15 depicts the sub-categories and groups for verification methods. Furthermore, Table 8 summarizes each group of methods separately alongside the works these groups are composed of.

4.4.1 Methods for detection of confidential information

Documents compose a source of information that must remain secret, under organizational ethics or privacy preservation laws, in order to prevent data leakages. Human experts have manually performed most of the efforts to tag sensitive or confidential information, hence being prone to workforce and time overheads, depending on the volume of documents to be analyzed. So deep neural networks have recently been employed in detecting these kinds of information automatically and quickly on large document collections. Neerbek et al. (2018) propose to learn phase structures discriminating between documents with sensitive and non-sensitive content using a recursive neural network (Irsoy and Cardie 2014) trained on labeled documents with no need to label every sentence itself. The authors argue that current keyword-based approaches for detecting sensitive information may fail to find complex sensitive information since these models do not take into account the sentences' context in the document. This architecture recursively receives a part of the input structure as input in each step. From sentences modeled as parse-trees, the neural network preserves the grammatical order of the sentences in a bottom-up manner, which consists of processing a node in the parse-tree in each step, ending at the root-node. A relative weighting strategy is implemented to distinguish between sensitive and non-sensitive information by higher weights to the kind of information detected in the sentence while computing a cross-entropy loss.

Measuring the harmfulness of information is a challenging task. In order to address it, Battaglia et al. (2020) defined a new data mining task called content sensitivity analysis which aims at assigning scores to data files taking into account their degree of sensitivity as a function $\hat{s} : \mathcal{O} \rightarrow [-1, 1]$, in which $\mathcal{O}$ is the domain of all user-generated contents. Therefore, given a user-generated object o'_i, $\hat{s}(o'_i) = 1$ if o'_i is maximally privacy-sensitive, on the contrary $\hat{s}(o'_i) = -1$. Since the notion of sensitive information is subjective depending on use cases and data protection regulations, $\hat{s} : \mathcal{O} \rightarrow [-1, 1]$ can be learned according to an annotated corpus of content objects satisfying:

$$\min \sum_{i=1}^{N} (\hat{s}(o'_i) - \beta_i^2), \tag{13}$$

Table 8 Summary of verification methods

Group	**Work**	**Neural models**	**PET**	***T***	***S***
Methods for DCI	Neerbek et al. (2018)	RecNN, GloVe	Relative weighting	*W*	SC
	Battaglia et al. (2020)	MLP, GloVe	CSA	*A*	SC
	Zhao et al. (2020)	ELMO, BERT, XL-Net, fasttext	Bias analysis	*W*	DC
	Tan and Celis (2019)	GloVe, ELMO, BERT, GPT, GPT-2	Bias analysis	*W*	LE
	Hutchinson et al. (2020)	BERT	Bias analysis	*W*	LE
	Gonen and Goldberg (2019)	Hard-Debiased, GN-GloVe, Glove	Bias analysis	*W*	TL
	Nissim et al. (2020)	word2vec	Analogy detection	*W*	LE
	Sweeney and Najafian (2019)	word2vec, GloVe, Conceptnet	Bias analysis by SS	*W*	LE
Methods for AA	Shrestha et al. (2017)	CNN	Feature extraction	*A*	DC
	Boumber et al. (2018)	CNN, word2vec, GloVe	Feature extraction	*A*	DC
	Barlas and Stamatatos (2020)	RNN, BERT, ELMO, GPT-2, ULMFiT	PTM	*A*	DC
	Caragea et al. (2019)	CNN, word2vec	Feature extraction	*A*	DC
Methods for AA of source code	Abuhamad et al. (2018)	LSTM+GRU	Feature extraction	*A*	OSC
	Abuhamad et al. (2019)	CNN, C&W	Feature extraction	*A*	OSC

Continues on next page...

Table 8 Summary of verification methods (Continued)

Group	**Work**	**Neural models**	**PET**	***T***	***S***
Compliance checkers	Song and Shmatikov (2019)	LSTM, seq2seq	Membership inference	*A*	DC
	May et al. (2019)	GloVe, InferSent, GenSent, USE, ELMO, GPT, BERT	Bias analysis	*W*	WCE
	Basta et al. (2020)	word2vec, ELMO	Bias analysis	*W*	WCE
	Vig et al. (2020)	GPT-2	Bias analysis	*W*	WCE
Methods for DDD	Song and Raghunathan (2020)	word2vec, GloVe, Fasttext, LSTM, Transformer, BERT, ALBERT	Disclosure attacks	*A*	PA
	Pan et al. (2020)	BERT, transformer-XL, XLNet, GPT, GPT-2, RoBERTa, XLM, ERNIE 2.0	Disclosure attacks	*W*	MR
	Carlini et al. (2019)	LSTM	Exposure metric	*W*	PA
	Akiti et al. (2020)	BiLSTM, GloVe, BERT, ELMO	SRL	*A*	SM

A stands for a set of protected attributes, **AA** stands for ‘authorship attribution’, **CSA** stands for ‘content sensitivity analysis’, **DC** stands for ‘document collection’, **DDD** stands for ’detection of data disclosures’, **DCI** stands for ‘detection of confidential information’, **GRU** stands for ‘Gated recurrent units’, **LE** stands for ‘language encoding’, **MR** stands for ‘medical records’, **OSC** stands for ‘open source contributors’, **PA** stands for ‘private attributes’, **PET** stands for ‘privacy-enhancing technology’, **PTM** stands for ‘pre-trained model’, ***S*** stands for ‘scenario’, **SC** stands for ‘sensitive content’, **SM** stands for ‘social media’, **SRL** stands for ‘sentence role labeling’, **SS** stands for ‘sentiment scoring’, ***T*** stands for ‘target’, **TL** stands for ‘transfer learning’, ***W*** stands for a set of target words, **WCE** stands for ‘word or context embeddings’.

in which $O' = \{(o'_i, \beta_i)\}$ is a set of N annotated objects $o'_i \in \mathcal{O}$ with their related sensitivity score β. In the experimental evaluation, the authors used GloVe embeddings to represent words and an MLP network to distinguish whether an input text is sensitive or not as a binary classification setting.

Discriminatory biases towards individuals or groups in DL models frequently arise from demographic attributes in the datasets leading to discriminatory decisions towards individuals or groups. Many recent approaches in the literature introduced methodologies for the detection and measurement of human biases in word representations. Zhao et al. (2020) quantify gender bias in language models for cross-lingual transfer, i.e., a scenario in which a model trained in one language is deployed to applications in another language. In the study, the authors evaluated three transfer learning methods, namely ELMO (Peters et al. 2018), BERT, and XL-Net (Yang et al. 2019), in addition to a modified fasttext model, applying a vector space alignment technique to reduce bias and a metric to quantify it. Tan and Celis (2019) analyze the extent to which contextualized word representation models (ELMO, BERT, GPT, and GPT-2) and GloVe embeddings can encode bias related to demographic attributes, such as gender, race, and intersectional identities. Hutchinson et al. (2020) reveal evidence of bias encoded by language models towards mentions of disabilities, focusing on the BERT model. Gonen and Goldberg (2019) conduct experiments hypothesizing that gender bias information might still be reflected in the distances between gender-neutral words in debiased embedding models, hence presenting the risk of recovery. Two embedding models are used by the authors, namely Hard-Debiased (Bolukbasi et al. 2016) and GN-GloVe, and then compared against the standard GloVe. Nissim et al. (2020) investigate the role analogies play in bias detection by considering these language structures as an inaccurate diagnosis for bias. In fact, the authors claim that analogies may have been overused, so possibly non-existing biases have been exacerbated, and others have been hidden. The word2vec model is used in the experiments alongside measures to detect analogies related to gender and profession names, such as gynecologist, nurse, and doctor.

Most of the approaches for measuring bias in word embedding models rely on distances in vector spaces. Claiming that insights based on those metrics are geometrically rich but limited with regard to model interpretability, Sweeney and Najafian (2019) present a framework for evaluating discriminatory biases on embedding models towards protected groups, such as national origin and religion. The authors introduce a novel metric for measuring fairness in word embedding models named RNSB, which takes into account the negative sentiment associated with terms related to demographic groups. In the study, three pre-trained word embedding models are evaluated, namely GloVe, word2vec, and ConceptNet (Speer et al. 2017). The last one was proved to be the least biased among the analyzed models.

4.4.2 Methods for authorship attribution (AA)

Determining the identity of written text's authors is a challenging problem with many real-world applications, such as plagiarism detection, spam filtering, phishing recognition, identification of harassers, text authenticity check, and detection of bot users on social media. AA is the task of distinguishing texts written by different authors based on stylometric features (Stamatatos 2009). Unveiling the authorship of written texts also poses privacy threats since both text content and the author's identity may be subject to data protection regulations. Furthermore, the author's willingness to share their identity is also a key factor to bear in mind during the development of AA applications which can be used, for instance, to re-identify anonymous texts or comments which are widely available on the internet as happened to the Netflix prize's dataset (Narayanan and Shmatikov 2008).

Addressing AA for short texts is an even more complex task compared to longer texts, according to Shrestha et al. (2017). The authors apply a CNN model over character n-grams, which capture patterns at the character level to help model the style of different authors. Firstly, the CNN architecture receives a sequence of character n-grams as input and forwards it across three modules: an embedding module, a convolutional module, and a fully connected softmax module. Secondly, the embedding module yields a vector representation for the character n-grams, passed over the convolutional layer to capture a feature map ω. Therefore, ω is pooled by max-over-pooling to produce $\tilde{v}_{k'}$ in the form

$$\tilde{v}_{k'} = \max_i \omega_{k'}[i], k' = 1, \ldots, m', \tag{14}$$

in which $\tilde{v}_{k'}$ is the maximum value in the k'-th feature map, and m' is the number of feature maps. Finally, after concatenating the pooled maps, the model generates a compacted text representation with the most important text features regardless of their original position. This representation is then input to a softmax layer that discriminates the text's author. The authors come up with the hypothesis that their model is able to learn features at morphological, lexical, and syntactical levels simultaneously.

Some scenarios impose further hurdles for AA models, as noticed in documents written by multiple authors whose detection resembles a multi-label classification problem. Boumber et al. (2018) design a CNN architecture for multi-label AA handling documents as sets of sentences that may present many labels. This model implements a strategy called collaborative section attribution, which consists of taking two possibilities into account concurrently. The first regards continuous sections written by a single author, while the second refers to the influence coauthors play on each other's writing style or, yet, editing passages written by others. Two word embedding models (word2vec and GloVe) are used on the multi-label AA architecture to represent the words in the documents. Like Shrestha et al. (2017), feature extraction steps have produced a vector to be input into a classification layer with a softmax activation function.

Divergences between training and test sets of texts represent another realistic and recurrent barrier for the task of AA. Barlas and Stamatatos (2020) take differences between textual genre and topic by dismissing information related to these two factors and solely concentrating on stylistic properties of texts associated with the personal writing styles of authors. In the work, the authors applied four pre-trained language models (i.e., ULMFiT, ELMO, GPT-2, and BERT) to cross-genre and cross-topic AA. Besides holding state-of-the-art results across an extensive range of NLP tasks, pre-trained models do not pose privacy for the text corpora used in their training steps since those corpora are typically composed of texts publicly available on the internet, such as news or Wikipedia articles. However, it is still unclear if the writing styles of the training data may affect the model behavior while performing AA.

Anonymity is occasionally preferred over explicit authorship information when a fair decision about a text must be made. For instance, during scientific papers' review, the author's name information in the manuscript up to evaluation would eventually bias the reviewers towards world-class authors over less renowned names. Caragea et al. (2019) investigate the effectiveness of deep neural networks for inferring the authorship information of scientific papers submitted to two top-tier NLP conferences (ACL and EMNLP). In order to perform their study, the authors implement a CNN model trained on scientific paper collections from the two conferences. The word2vec embedding model produces the representations for the words in the documents prior to inputting these vectors to the convolutional layers. Separate sub-components of the CNN are responsible for feature extraction from paper content, stylometric features, and the entries on each paper's references. On the network top, a fully connected layer and a classification layer with a softmax activation function predict the class for each article after receiving the outputs from the three sub-components for feature extraction.

4.4.3 Methods for authorship attribution of source code

AA is an NLP task regarding the assignment of correct authors to contentious samples of work whose writing is anonymous or disputable, including source code in programming languages (Burrows et al. 2014). Abuhamad et al. (2018) argue that AA of source code poses privacy threats regarding developers working on open source projects when they refuse to reveal their identities. However, it can enhance applications for digital forensics, plagiarism detection, and identification of malware code developers. So the authors propose a DL-based system for AA of source code relying on an LSTM model with gated recurrent units for yielding representations of TF-IDF vectors of code files. A classifier receives the features learned by the LSTM architecture and performs AA, and technical analyses are conducted. In a similar fashion, Abuhamad et al. (2019) come up with CNN models for this task. Both TF-IDF and C&W (Collobert et al. 2011) vectors generate representations for source code, as well as are compared with respect to the final classification performance. Therefore, the neural networks learn features and classify the input vectors by assigning their authorship.

4.4.4 Compliance checkers

Data privacy has been assured by laws in many countries, such as the EU's GDPR, hence some approaches in the literature of privacy-preserving NLP verify whether DL models complied or not with such regulations or with concepts of fairness. It is a common practice to use user-generated data to input language models for applications like word prediction, automatic question-answering, and dialogue generation. However, there should be transparency in the data collection and utilization. Song and Shmatikov (2019) develop an auditing technique that is able to inform users if their data was used during the training step of text generation models built on LSTMs. This study also analyzes to what extent language models memorize their training data since it can be considered a problem for both NLP and privacy aspects. According to the authors, auditing can be thought of as a membership inference towards a model at the user level.

The detection of implicit human biases encoded by word representation models is an extensive field of research in NLP. Such biases can be propagated into downstream applications and lead to discriminatory decisions, hence breaching good practice protocols and regulations that enforce fairness. Thus, we include in the group of compliance checkers works that propose new methods for detecting bias in embeddings. Here, the major focus consists of bias detection, contrary to Section 4.2.5, which focuses on the bias reduction and mitigation. Common means for bias identification are tests like the Word Embedding Association Test (WEAT) (Caliskan et al. 2017), which measures the association between two equally sized sets of target word embeddings $\tilde{V}$ and $\tilde{W}$, and two sets of attribute word embeddings $\tilde{A}$ and

$\tilde{B}$. WEAT's test statistic is computed in the form:

$$\rho(\tilde{V}, \tilde{W}, \tilde{A}, \tilde{B}) = [\sum_{\tilde{v}\in\tilde{V}} \rho(\tilde{v}, \tilde{A}, \tilde{B}) - \sum_{\tilde{w}\in\tilde{W}} \rho(\tilde{w}, \tilde{A}, \tilde{B})], \tag{15}$$

in which $\rho(\tilde{t}, \tilde{A}, \tilde{B})$ is the difference between the mean cosine similarity of the attribute word embeddings $\tilde{a}$ and $\tilde{b}$, respectively in $\tilde{A}$ and $\tilde{B}$, (May et al. 2019) computed as

$$\rho(\tilde{t}, \tilde{A}, \tilde{B}) = [mean_{\tilde{a}\in\tilde{A}} cos(\tilde{t}, \tilde{a}) - mean_{\tilde{b}\in\tilde{B}} cos(\tilde{t}, \tilde{b})] \tag{16}$$

Specifically, $\rho(\tilde{t}, \tilde{A}, \tilde{B})$ is a measure of how associated a target word embedding $\tilde{t}$ and a attribute word embeddings are, while $\rho(\tilde{V}, \tilde{W}, \tilde{A}, \tilde{B})$ is a measure of the association between the sets of target word embeddings and the attribute ones (Caliskan et al. 2017). Additionally, permutation test on $\rho(\tilde{V}, \tilde{W}, \tilde{A}, \tilde{B})$ computes the significance of the association between both pairs of target and attribute word embeddings (May et al. 2019), as

$$\tilde{p} = Pr_i[\rho(\tilde{V}_i, \tilde{W}_i, \tilde{A}, \tilde{B}) > \rho(\tilde{V}, \tilde{W}, \tilde{A}, \tilde{B})], \tag{17}$$

in which $(\tilde{V}_i, \tilde{W}_i)$ stand for all partitions in $\tilde{V} \cup \tilde{W}$, and $\tilde{p}$ is one sided p-value of the test. Finally, the magnitude of the association (Caliskan et al. 2017; May et al. 2019) is calculated by

$$\mu = \frac{mean_{\tilde{v}\in\tilde{V}} \rho(\tilde{v}, \tilde{A}, \tilde{B}) - mean_{\tilde{w}\in\tilde{W}} \rho(\tilde{w}, \tilde{A}, \tilde{B})}{std\_dev_{\tilde{t}\in\tilde{V}\cup\tilde{W}} \rho(\tilde{t}, \tilde{A}, \tilde{B})}. \tag{18}$$

Since WEAT is meant to detect biases at the word level, May et al. (2019) propose a generalized version of this test titled Sentence Encoder Association Test (SEAT) to uncover biases at the phrase and sentence levels. To do so, SEAT is inputted with sets of contextualized embeddings for sentences with similar structure, focusing on specific words, like ethnic names, and attributes that are related to biases towards the specific words. The experiments performed by the authors targeted two types of bias, namely the black woman stereotype (Harris-Perry 2011) and double binds defined as antagonistic expectations of femininity and masculinity (Harris-Perry 2011; May et al. 2019). A total of seven sentence encoders (Table 8) are analyzed with SEAT, drawing evidence that such encoders evince less bias than prior word embedding models.

Basta et al. (2020) evaluate language models for both English and Spanish languages with regard to gender bias. While the former language does not present distinct gendered forms for most nouns, the latter heavily does. Furthermore, the authors do not use WEAT or SEAT as methods for detecting biases on ELMO and word2vec embeddings. Their experimental evaluation takes into account metrics based on principal component analysis, gender direction, clustering, and supervised classifiers, namely support vector machines and K-NN. Similarly, Vig et al. (2020) present a methodology relying on causal mediation analysis (Pearl 2001) to analyze which internal components of pre-trained GPT-2 models concentrate most of the gender bias learned from the training data. Causal mediation analysis gauges the extent network neurons mediate gender bias individually and jointly. The study concludes that gender bias is concentrated in a few language model components, and the individual effects of some components may be amplified by interactions. Last, the authors also argue that the total gender bias effect approximates the sum of both direct and indirect effects related to the information flows from input to output variables.

4.4.5 Methods for detection of data disclosures

Language models often handle private or sensitive attributes, such as demographic information or stylometric features, which should remain unveiled for parties using or accessing these neural networks. In order to verify the susceptibility of such models to breaching private data, recent studies have come up with attacking approaches or information tracking methodologies. Song and Raghunathan (2020) develop three classes of attacks to study which kinds of information could be leaked from word embedding models. First, embedding inversion attacks aim to invert existing vector representations to their raw text formats, including private content. Second, attribute inference attacks check how likely embedding models are to reveal sensitive attributes from the training data. Finally, membership inference attacks demonstrate the ability to recover training samples when an adversarial has access to both the language model and its outputs.

Pan et al. (2020) investigate how 8 general-purpose language models (Table 8) capture sensitive information which can later be disclosed by adversaries for harassment afterward. The authors design two classes of attacks aiming at disclosing sensitive information from the tested models. First, pattern reconstruction attacks enable adversaries to recover sensitive segments of sequences, such as genome sequences, used for model training. Second, keyword inference attacks target sensitive keywords in unknown texts, like medical descriptions. The study has found that leaked embeddings present a high potential to allow adversaries to disclose sensitive information from users.

Unintended memorization is a noticeable drawback of neural networks that may put training instances' privacy at risk of disclosure in case these instances hold unique or rare values, such as IDs, addresses, or credit card numbers. Aiming to assess this problem, Carlini et al. (2019) describe a testing methodology that limits data exposure by minimizing its memorization. An exposure metric is proposed by the authors to quantify the propensity of data disclosure by neural models. Consequently, the study finds that memorization may not be due to excessive model training but a side-effect that appears at the early stages of training and prevails on several models and training frameworks.

On social media platforms, such as Twitter, Facebook, and Instagram, people share huge amounts of content every day, often including their own private information, unaware of the privacy risks these actions may bring about. Akiti et al. (2020) detect emotional and informational self-disclosure on Reddit data using semantic role labeling, which is a technique for recognizing predicate-argument pairs in sentences. According to the authors, emotional self-disclosures regard the user's feelings about people or things, while the informational ones are related to revealing personal information, such as age, career status, address, and location. As a result, the authors overtake state-of-the-art methods, demonstrating the approach's efficiency in detecting personal disclosures.

5 Applications and datasets

Due to the growing literature on privacy-preserving NLP, we list the benchmark data used for the experiments in the reviewed works in Table 9. We also include NLP tasks in the table to point out the utility of the proposed privacy-preserving methods.

Many NLP tasks are performed over private data, as sentiment analysis (Feyisetan et al. 2020; Lyu et al. 2020; Li et al. 2018), AA (Boumber et al. 2018), and neural machine translation (Feng et al. 2020), hence putting privacy at risk from disclosures or attacks. PETs, such as FHE, DP, adversarial learning, and FL, are potential solutions for such privacy issues. However, one should find a balance for the privacy-utility tradeoffs for each one of these technologies in a different manner. So the performance requirements of an NLP task, alongside the computational power of the devices in the learning scenario, will play a significant role in choosing suitable PETs. For instance, FHE may lead to memory overheads that are hard to manage for small devices like smartphones, slowing computations down (Huang et al. 2020). Therefore, a lighter encryption scheme or even controlled noise by DP would be preferred in such a scenario.

Although Table 9 presents a large number of entries, we can notice that the lack of data for privacy-preserving NLP tasks is still an open problem since the number of datasets per task is small for most of the tasks. Firstly, this data availability issue is mainly related to the hardness of annotating sensitive text content (Eder et al. 2020) and human biases (Zhao et al. 2020). Secondly, another hurdle to generating data for privacy purposes in NLP regards the safe release of PHI documents (Melamud and Shivade 2019), such as clinical notes and electronic health records, which is sometimes prohibited by legal terms. Finally, there are further problems regarding the availability of datasets for languages other than English, yet a frequent situation across NLP tasks.

6 Privacy metrics

Privacy metrics aim at gauging the amount of privacy that users of a system experience and the extent of protection privacy-preserving methods provide (Wagner and Eckhoff 2018). In the privacy-preserving NLP domain, many privacy metrics have been proposed in recent years. Table 10 shows ten privacy metrics we have identified in the surveyed works. The table presents metric names, alongside their privacy-related target that represents the privacy-related guarantee to be measured, as the privacy budget for DP or the amount of bias in word representations. Therefore, we summarize these metrics as follows. For detailed mathematical notation, we advise the reader to refer to the papers cited along column 'Work'.

- *ϵ-DP*. In DP (Dwork 2008), the parameter ϵ is the upper bound to the probability of output to be changed by the addition or removal of a data instance (Dwork et al. 2006; Andrew et al. 2019). This parameter is related to the privacy budget, which regards the amount of privacy to be protected. The closer the values of ϵ are to zero, the better the privacy protection. DP is broadly used for privacy-preserving NLP methods, such as FL (Zhu et al. 2020), authorship obfuscation (Fernandes et al. 2019), and representation learning (Feyisetan et al. 2020; Lyu et al. 2020).
- *inBIAS*. Gender bias is a broadly researched privacy issue in word and language representations. Thus, metrics to quantify gender bias in embeddings have been proposed. InBias (Zhao et al. 2020) measures the intrinsic gender bias in multilingual word embeddings from a word-level perspective. It is based on distances between gendered words, like occupations, and gendered seed words, like gender pronouns. For instance, if

Table 9 Summary of tasks and benchmark datasets for privacy-preserving NLP

NLP task	**Datasets & works**
Anonymization of document images	Invoice images dataset (Sánchez et al. 2018).
Authorship attribution	20-author set, 50-author set (Fernandes et al. 2019), Twitter (Shrestha et al. 2017), MLPA-400, PAN-2012 (Boumber et al. 2018), ACL papers, EMNLP papers (Caragea et al. 2019), Enron (Feyisetan et al. 2020), CMCC (Barlas and Stamatatos 2020).
AA of source code	GitHub, Google Code Jam (2008-2016) (Abuhamad et al. 2018, 2019).
Author obfuscation	PAN11, PAN12 (Feyisetan et al. 2019).
Bias assessment	Winobias (Vig et al. 2020; Tan and Celis 2019), Winogender (Vig et al. 2020), MIBs (Zhao et al. 2020), 1BWord, BookCorpus (Tan and Celis 2019), Wikipedia, WebText (Tan and Celis 2019), Reddit (Hutchinson et al. 2020), Professions (Gonen and Goldberg 2019), Word2vec test set (Nissim et al. 2020), Sentiment training (Speer et al. 2017).
Biomedical translation	Biomed (Basta et al. 2020).
Capitalization	MedText (Melamud and Shivade 2019).

Continues on next page...

Table 9 Summary of tasks and benchmark datasets for privacy-preserving NLP (Continued)

NLP task	**Datasets & works**
Clinical notes generation	MedText-2 Melamud and Shivade (2019), MedText-103, WikiText-2, WikiText-103 (Melamud and Shivade 2019).
Content sensitive analysis	Dataset created by the authors (Battaglia et al. 2020).
Co-reference resolution	Ontonotes 5.0, WinoBias (Zhao et al. 2018).
De-identification of medical records	2014 i2b2 (Liu et al. 2017b; Dernoncourt et al. 2017; Friedrich et al. 2019), 2016 N-GRID (Liu et al. 2017b), MIMIC (Dernoncourt et al. 2017).
Detection of altered mental status	AMS (Obeid et al. 2019).
Detection of demographic biases	CF Twitter, 100 Authors Twitter (Barrett et al. 2019), PAN14 Blogs, PAN14 Reviews (Barrett et al. 2019), PAN14 SoMe, PAN16 Rand (Barrett et al. 2019), German Facebook comments (Papakyriakopoulos et al. 2020), Jigsaw dataset, Multilingual Tweets (Gencoglu 2020), WikiDetox dataset, Gab Hate Corpus (Gencoglu 2020).
Detection of self-disclosures	Reddit (Akiti et al. 2020).
Detection of sensitive information	Justice corpus, Healthcare corpus (Martinelli et al. 2020), Enron (Neerbek et al. 2018).

Continues on next page...

Table 9 Summary of tasks and benchmark datasets for privacy-preserving NLP (Continued)

NLP task	**Datasets & works**
Dialog generation	Cornell movie dialogs, Ubuntu dialogs (Song and Shmatikov 2019).
Disentangling latent spaces	Yelp, Amazon reviews (John et al. 2019).
Evaluation of privacy guarantees	GloVe vocabulary (Feyisetan et al. 2019).
Evaluation of word vector distortion	WordSim353 (Sweeney and Najafian 2020).
Extraction of sensitive information	Randomly generated citizen IDs, Genome (Pan et al. 2020).
Feature extraction	Louisiana Tumor Registry (Alawad et al. 2020), Kentucky Cancer Registry (Alawad et al. 2020).
Keyword inference attacks	Airline reviews, CMS public healthcare records (Pan et al. 2020).
Keyword search	Amazon reviews, Enron, Science dataset (Liu and Wang 2020), 20 news groups (Dai et al. 2019).
Medical document anonymization	MEDDOCAN, NUBES (Pablos et al. 2020).
Memorization detection	Enron (Carlini et al. 2019).
Model training	Penn Treebank (PTB), WikiText-103 (Carlini et al. 2019).
Native language detection	L2-Reddit, TOEFL17 (Kumar et al. 2019).

Continues on next page...

Table 9 Summary of tasks and benchmark datasets for privacy-preserving NLP (Continued)

NLP task	**Datasets & works**
Natural language inference	MedNLI (Melamud and Shivade 2019), SICK-E (Feyisetan et al. 2019).
Natural language understanding	GLUE (Huang et al. 2020).
Neural machine translation	WMT (Feng et al. 2020; Basta et al. 2020), SATED (Song and Shmatikov 2019), Europarl (Basta et al. 2020; Song and Shmatikov 2019), TEDx, WMT13 (Spanish) (Basta et al. 2020), Newstest2012, Newstest2013 (Font and Costa-jussà 2019).
News recommendation	Adressa, MSN-News (Qi et al. 2020).
Next word prediction	en_US keyboard clients (Chen et al. 2019), pt_BR keyboard clients (Chen et al. 2019), Reddit posts (McMahan et al. 2018), Logs data, Cache data (Hard et al. 2018).
Opinion polarity classification	MPQA (Feyisetan et al. 2019).
Paraphrase detection	MRPC (Feyisetan et al. 2019).
Pos-tagging	TrustPilot, WebEng, AAVE (Li et al. 2018).
Prediction adaptation	Amazon reviews, 20 news groups (Clinchant et al. 2016).
Privacy audit	Search logs (Feyisetan et al. 2020).

Continues on next page...

Table 9 Summary of tasks and benchmark datasets for privacy-preserving NLP (Continued)

NLP task	**Datasets & works**
Privacy-aware text rewriting	Yelp, Facebook comments, DIAL (Xu et al. 2019).
Question answering	InsuranceQA (Feyisetan et al. 2020).
Question type classification	TREC-6 (Feyisetan et al. 2019).
Recognition of private information	CODE ALLTAG$_{S+d}$, CODE ALLTAG$_{XL}$ (Eder et al. 2020).
Sentence intent classification	TREC (Zhu et al. 2020).
Sentiment analysis	IMDB movie reviews (Feyisetan et al. 2020), Trustpilot (Coavoux et al. 2018; Mosallanezhad et al. 2019), Trustpilot (Lyu et al. 2020; Li et al. 2018).
Sentiment bias evaluation	Identity terms (Sweeney and Najafian 2020).
Sentiment prediction	Dialectal tweets (DIAL) (Elazar and Goldberg 2018), MR, CR, SST-5 (Feyisetan et al. 2019).
Sentiment valence regression	SemEval-2018 Task 1 (Sweeney and Najafian 2020).
Stylometrics	BookCorpus (Song and Raghunathan 2020).
Synthesized language generation	Twitter posts (Oak et al. 2016).
Text style transfer	Yelp, Amazon reviews (John et al. 2019).

Continues on next page...

Table 9 Summary of tasks and benchmark datasets for privacy-preserving NLP (Continued)

NLP task	**Datasets & works**
Topic classification	AG news (Coavoux et al. 2018; Lyu et al. 2020), Deutsche Welle (Coavoux et al. 2018), Blog authorship (Coavoux et al. 2018; Lyu et al. 2020), Document set (Fernandes et al. 2019).
Toxicity classification	Wikipedia Talk Sweeney and Najafian (2020).
Tweet-mention prediction	Dialectal tweets (DIAL) (Elazar and Goldberg 2018), PAN16 (Elazar and Goldberg 2018; Barrett et al. 2019).
Word analogy	SemBias, Google Analogy, MSR (Zhao et al. 2018; Kaneko and Bollegala 2019), SemEval (Kaneko and Bollegala 2019).
Word embedding model training	Wikipedia (Song and Raghunathan 2020; Zhao et al. 2018) Wikipedia, Word lists (Kaneko and Bollegala 2019), Google News (Bolukbasi et al. 2016), Mixed source sentences (Font and Costa-jussà 2019), Wikipedia, German Tweets, German Facebook (Papakyriakopoulos et al. 2020).
Word extraction	Sentiment Lexicon (Sweeney and Najafian 2020).
Word prediction	Reddit, Wikitext-103 (Song and Shmatikov 2019).
Word similarity	WS353, RG-65, MTurk, RW, MEN (Zhao et al. 2018; Kaneko and Bollegala 2019), SimLex (Kaneko and Bollegala 2019), STS14 (Feyisetan et al. 2019).

Table 10 Summary of metrics for privacy in NLP

Metric	Privacy-related target	Work
ϵ-DP	Privacy budget	Fernandes et al. (2019) Feyisetan et al. (2020) Zhu et al. (2020) Lyu et al. (2020)
inBias	Bias in multi-lingual embeddings	Zhao et al. (2020)
RNSB	Unintended demographic bias	Sweeney and Najafian (2019)
$Exposure$	Propensity for revealing data	Carlini et al. (2019)
Entropy	Sensitive information leakage	Xu et al. (2019)
P-Acc	Prediction of sensitive attributes	Xu et al. (2019)
M-Acc	Label probabilities for sentences	Xu et al. (2019)
FNED	Unintended biases	Gencoglu (2020)
FPED	Unintended biases	Gencoglu (2020)
S-PDTP	Prediction of private records	Melamud and Shivade (2019)

a feminine occupation presents a larger distance to the feminine seed gendered word when compared to the distance between its equivalent masculine words, it can be seen as a sign of bias. Furthermore, this metric presents the advantage of enabling bias evaluation for embeddings of words in languages other than English.

- *RNSB*. Bias towards demographic attributes (e.g., national origin and religion) can be gauged by this measure (Sweeney and Najafian 2019). It works by measuring the association of positive and negative sentiments and the protected demographic attributes in word embedding models.
- $Exposure$. DL models may memorize training data, thereby putting data privacy at risk if an attacker tries to recover the original training samples. Therefore, the metric of $exposure$ measures the unintended memorization of unique or rare data instances (Carlini et al. 2019). It can be used during the model training to evaluate the risks of a successful attack.
- *Entropy*. In the task of privacy-aware text rewriting, the generated text, which does not contain sensitive information, can be evaluated by the metric of entropy (Xu et al. 2019). Given a classifier that predicts the probability of sensitive attributes in the generated sentences in this task, higher entropy in the predictions is a sign of low risk of information leakage.
- *P-Acc*. This metric measures the ratio of correct predictions of the sensitive attribute obfuscated in the task of privacy-aware text rewriting (Xu et al. 2019). Therefore, lower values of this metric suggest better obfuscation.
- *M-Acc*. This is another metric to evaluate the privacy of a privacy-preserving text re-writing task. It compares the accuracies of predicting the labels for both original and generated sentences. Therefore, the approval for the generated sentence is based on the drop in the probability of the protected attributed after the re-writing task (Xu et al. 2019).
- *FNED* and *FPED*. These two metrics evaluate unintended biases towards demographic attributes, such as gender, religion, nationality, and language (Gencoglu 2020). These two measures are, based on the false negative and false positive ratios of classification tasks like cyberbullying detection. The total amount of unintended bias of a model is assumed to be the sum of both measures (Gencoglu 2020).
- *S-PDTP*. This metric measures the privacy protection for documents, like clinical notes, which should not be shared without PETs, e.g., DP or de-identification. S-PDTP (Melamud and Shivade 2019) estimates the privacy risks for synthetic clinical notes generated from private ones. Therefore, the lower the scores of S-PDTP are, the less private information from the original clinical notes is predicted.

7 Discussion and open challenges

Protecting NLP data against privacy-related issues presents hurdles as to utility task goals, computation scenarios, DL architectures, and the properties of the datasets to be used. For instance, the removal of protected terms from the vocabulary of a language model may disrupt the semantics of the remaining word representations and compromise the utility of such a privacy-preserving method. In this section, we discuss the challenges of integrating privacy protection into NLP models regarding five aspects: traceability, computational overhead, dataset size, bias prevalence in embeddings, and privacy-utility tradeoff. We also point out suggestions for future directions in privacy-preserving NLP.

7.1 Traceability

Obfuscating private text attributes is a tricky problem since these attributes may not be explicitly presented in the text. Taking the problem of gender disclosure as an example, the gender of an author can be inferred from sentences, even though no explicit mention of their actual gender is made in the text. Other private attributes can also be easily inferred from the text, such as location, age, native language, and nationality. This issue of getting rid of private information is hardened by more complex text data, such as e-mails, documents, clinical notes, or electronic medical records, which are subject to data protection regulations, such as the HIPAA or the EU's GDPR. Such regulations hinder personal or medical data sharing, or public release, without safeguarding methods, such as sanitization, de-identification, DP, or encryption. However, when it comes to privacy guarantees, these methods present pitfalls whose solution still has room for new contributions, like diminishing the traceability of the original data.

Given an adversary that accesses a de-identified or anonymized document, it should not be able to trace the private document version. First, data sanitization fails at bringing formal guarantees that traceability is unpractical for an adversary. Second, de-identification replaces the private attributes with generic surrogate terms. However, in case these terms do not belong to the same semantic category as the original ones, the document context is disrupted. Third, traceability is also a problem for representation learning since private attributes can be recovered from learned latent representations of words and documents, and implicit private information can be found on the gradients of DL models (Qi et al. 2020). Further, searchable encryption is an efficient method against traceability since it forces adversaries to expend high computing power to beak ciphertexts, despite threats of file injection attacks. Finally, DP introduces formal privacy guarantees against this issue (Obeid et al. 2019), enforcing the principle of indistinguishability between computation outputs. We believe that more efforts toward DP will push the boundaries against data traceability in the future.

7.2 Computation overhead

Privacy preservation often comes at the cost of computation overheads related to model run-time, memory footprint, and bandwidth consumption. Infrastructure requirements represent a key point for the development of privacy-preserving models since the computation scenario is usually constrained. For instance, some mobile devices may feature poor memory and processing power in a distributed setting. So implementing memory-consuming FHE or MPC schemes may be prohibitive. MPC overhead related to bandwidth absorption (Feng et al. 2020) is a promising problem to work towards a solution in the future since distributed computation scenarios are increasing at the pace that mobile devices become increasingly popular. Additionally, noise control for FHE on DL networks is another challenging issue. FHE works by adding noise terms to ciphertexts, which will grow with mathematical operations of addition and multiplication. When a DL model (e.g., LSTM, BERT, and ELMO) presents recurrence functions, the level of multiplicative depth can increase the level of noise, corrupting the model outputs once they are decrypted (Al Badawi et al. 2020). Furthermore, MPC and FHE may slow down computations on massive databases. DP noise also influences the model run-time but with the advantage of trading privacy protection with computation overhead. So the amount of protected privacy can be accounted for as a budget. Finally, an important open challenge for FL regards coming up with models that do not slow down the application run-time. The importance of addressing this challenge is evinced by real-time mobile applications, like virtual smartphone keyboards, that should ideally provide instant results for users. More than privacy protection, the solution for this problem influences the application's usability and rating by users.

7.3 Dataset size

Dataset size plays an important role in the decision for a PET in real-world NLP applications. The importance of this factor is based on its influence on the generalization power of adversarial learning and FL and the privacy budget of DP. For adversarial learning, the efficiency of the adversarial classifier that unlearns a private attribute can be negatively affected if the dataset is overly small. Similarly, the dataset size can be a burden for FL since it may not be homogeneously distributed across the devices in the federated setting. For instance, the storage size of distributed devices, such as smartphones or internet of things sensors, can present different storage sizes. Therefore, an FL method must consider this point before designing the aggregation function on the global model and integrating additional PETs, like FHE or DP, into the learning setting.

In FL, sometimes the small number of distributed devices is also a burden to the model performance, e.g., data of a small number of users is insufficient to train accurate news recommendation models (Qi et al. 2020). Although

DP guarantees hold on smaller datasets, the level of noise injected into the data may occasionally vary. Further, DP noise can be input into model gradients instead of data instances themselves (Abadi et al. 2016). When it comes to combining DP with FL, it is a good practice to define the privacy budget locally based on each distributed device. Establishing this parameter in the global model without considering the differences in data storage between the devices in the learning setting may lead to a scenario in which privacy is protected for devices with smaller amounts of data. In contrast, devices storing large amounts of data would not enjoy enough protection. We believe many advances can still be made with respect to the influence of dataset sizes in privacy-preserving NLP, especially in backdrops with many devices featuring different data storage capacities.

7.4 Bias prevalence in embeddings

Bias and fairness are two critical privacy-related topics in NLP due to the ethical consequences they cause on automatic decision-making. These topics thereby influence the daily lives of a large number of people since many real-word systems are based on word and sentence embedding models, such as word2vec, GloVe, fasttext, and BERT, which were found encoding and propagating discriminatory human biases towards demographic attributes, especially gender, race, nationality, and religion, on to downstream applications. As an example, the machine translation task was found to output masculine words for non-gender-specific English words, like friend, when a feminine word was expected in the translated sentence (Font and Costa-jussà 2019). Reducing or removing human biases from embedding models mainly relies on adversarial training. However, adversarial debiasing is not always able to remove biases completely (Elazar and Goldberg 2018; Barrett et al. 2019). Therefore, bias prevalence in debiased embeddings is an open challenge.

Additionally, the evaluation step for debiasing and fairness solutions often does not include a human evaluation round. We believe that inviting people for such an evaluation, especially those individuals from groups to which biases are often directed, would enhance the quality of the debiased embeddings and their successive applications. Approaches for reducing discriminatory biases based on statistical dependency, namely causality, are also potential solutions for this problem (Vig et al. 2020). This research direction may also explore the cases when biases arise from data sampling methods, such as sampling data instances mostly related to a specific sensitive attribute, whereas others are left out. Finally, the gap between PETs and debiased embedding models must still be bridged. PETs protect the data content during model computations, but the predictions on these protected datasets may still be biased. Thus, there is an ever-increasing need for privacy-preserving fair NLP methods.

7.5 Privacy-utility tradeoff

In the reviewed works, it is a recurrent statement that privacy preservation in NLP is exchanged for performance on NLP tasks. Many PETs, like DP, AL, and data de-identification, reduce the model utility, whereas data privacy is protected. For instance, the usability of medical text is frequently reduced by de-identification techniques (Obeid et al. 2019). Similarly, the context of documents can be lost by replacing sensitive terms with surrogate words (Eder et al. 2019). In embedding models, the problem of semantic disruption may be caused by DP noise, which is injected into the representations, aiming to obfuscate the text's private attributes. Therefore, the final results of downstream applications can be compromised.

When a PET is implemented alongside the DL model for NLP, the model itself may face drawbacks. For instance, searches on encrypted documents can be less efficient for the users than searches on the plaintext. DP enables finding a good balance for the privacy-utility tradeoff (Abadi et al. 2016; Huang et al. 2020) by reducing the accuracy losses but still preserving privacy to some extent. FL is an effective privacy-preserving method, but as long as no data is exchanged between the distributed devices and the global model, the aggregation strategy to update the global model plays an important role in this tradeoff. Some distributed devices in a federated setting may influence the global model to a larger degree based on their larger locally stored data. Additionally, devices with poor bandwidth can be left out of model updates. So the local performances can drop as well. Therefore, balancing this tradeoff is certainly a key factor in designing privacy-preserving methods for NLP.

8 Conclusion

This article presents the first extensive systematic review of DL methods for privacy-preserving NLP, covering a large number of tasks, PETs, privacy issues, and metrics for evaluating data privacy. We propose a taxonomy structure to classify the existing works in this field as a guide to readers, with the advantage of enabling this structure to be extended to future works and regular machine learning methods. Our review covers over sixty DL methods for privacy-preserving NLP published between 2016 and 2020, which safeguard privacy from the perspectives of data, models, computation scenarios, and PETs. These methods explore the privacy-utility tradeoff to balance privacy protection and NLP task performance based on the assumption that privacy guarantees often come at a performance cost.

To sum up, we investigated the question of how to keep text private. Firstly, we approached this endeavor by considering the data properties, such as type, size, and content. Secondly, we describe the factors that influence choosing a suitable PET to be implemented with the DL model. Thirdly, our extensive benchmark datasets and privacy measures tables can assist researchers and practitioners in successive works. Finally, the reviewed works show that PETs will play an ever-increasing role in future NLP applications since standard solutions may not hinder privacy threats when processing text data.

9 Acknowledgments

This work was supported by the EU's Horizon 2020 project TRUSTS under grant agreement No. 871481. Know-Center is funded within the Austrian COMET Program – Competence Centers for Excellent Technologies – under the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Economy, Family and Youth, and the State of Styria. COMET is managed by the Austrian Research Promotion Agency (FFG).

Statements and declarations

All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

References

Alan F Westin. Privacy and freedom. *Washington and Lee Law Review*, 25(1):166, 1968.

European Commission. Reform of eu data protection rules. `https://ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes_en.pdf`, 2018. Date: 2018-05-25.

GW Van Blarkom, John J Borking, and JG Eddy Olk. Handbook of privacy and privacy-enhancing technologies. *Privacy Incorporated Software Agent (PISA) Consortium, The Hague*, 198:14, 2003.

Kyoohyung Han, Seungwan Hong, Jung Hee Cheon, and Daejun Park. Logistic regression on homomorphic encrypted data at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9466–9471, 2019. doi: https://doi.org/10.1609/aaai.v33i01.33019466.

Elisabeth Eder, Ulrike Krieg-Holz, and Udo Hahn. De-identification of emails: Pseudonymizing privacy-sensitive data in a german email corpus. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 259–269, 2019. doi: https://doi.org/10.26615/978-954-452-056-4_030.

Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association*, 24(3):596–606, 2017. doi: https://doi.org/10.1093/jamia/ocw156.

Su Lin Blodgett and Brendan O'Connor. Racial disparity in natural language processing: A case study of social media african-american english. *arXiv preprint arXiv:1707.00061*, 2017.

Joni Salminen, Rohan Gurunandan Rao, Soon-gyo Jung, Shammur A Chowdhury, and Bernard J Jansen. Enriching social media personas with personality traits: A deep learning approach using the big five classes. In *International Conference on Human-Computer Interaction*, pages 101–120. Springer, 2020. doi: https://doi.org/10.1007/978-3-030-50334-5_7.

Zecheng He, Tianwei Zhang, and Ruby B Lee. Model inversion attacks against collaborative inference. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 148–162, 2019. doi: https://doi.org/10.1145/3359789.3359824.

Ángel Sánchez, José F Vélez, Javier Sánchez, and A Belén Moreno. Automatic anonymization of printed-text document images. In *International Conference on Image and Signal Processing*, pages 145–152. Springer, 2018. doi: https://doi.org/10.1007/978-3-319-94211-7_17.

Mohammed Alawad, Hong-Jun Yoon, Shang Gao, Brent Mumphrey, Xiao-Cheng Wu, Eric B Durbin, Jong Cheol Jeong, Isaac Hands, David Rust, Linda Coyle, et al. Privacy-preserving deep learning nlp models for cancer registries. *IEEE Transactions on Emerging Topics in Computing*, 2020. doi: https://doi.org/10.1109/TETC.2020.2983404.

Yitong Li, Timothy Baldwin, and Trevor Cohn. Towards robust and privacy-preserving text representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 25–30, 2018.

Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. Privacy risks of general-purpose language models. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1314–1331. IEEE, 2020. doi: https://doi.org/10.1109/SP40000.2020.00095.

Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 619–633, 2018. doi: https://doi.org/10.1145/3243734.3243834.

Amine Boulemtafes, Abdelouahid Derhab, and Yacine Challal. A review of privacy-preserving techniques for deep learning. *Neurocomputing*, 384:21–45, 2020. doi: https://doi.org/10.1016/j.neucom.2019.11.041.

Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178, 2009. doi: https://doi.org/10.1145/1536414.1536440.

Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

Bo Yin, Hao Yin, Yulei Wu, and Zexun Jiang. Fdc: A secure federated deep learning mechanism for data collaborations in the internet of things. *IEEE Internet of Things Journal*, 2020. doi: https://doi.org/10.1109/JIOT.2020.2966778.

Oded Goldreich. Secure multi-party computation. *Manuscript. Preliminary version*, 78, 1998.

Weiwei Jia, Haojin Zhu, Zhenfu Cao, Xiaolei Dong, and Chengxin Xiao. Human-factor-aware privacy-preserving aggregation in smart grid. *IEEE Systems Journal*, 8(2):598–607, 2013. doi: https://doi.org/10.1109/JSYST.2013.2260937.

Shervin Minaee and Zhu Liu. Automatic question-answering using a deep similarity neural network. In *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 923–927. IEEE, 2017. doi: https://doi.org/10.1109/GlobalSIP.2017.8309095.

Eliyahu Kiperwasser and Yoav Goldberg. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327, 2016.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, et al. Tensor2tensor for neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 193–199, 2018.

Ruhi Sarikaya, Geoffrey E Hinton, and Anoop Deoras. Application of deep belief networks for natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4):778–784, 2014. doi: https://doi.org/10.1109/TASLP.2014.2303296.

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 115–124, 2017a. doi: https://doi.org/10.1145/3077136.3080834.

Samuel Sousa, Evangelos Milios, and Lilian Berton. Word sense disambiguation: an evaluation study of semi-supervised approaches withword embeddings. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020. doi: https://doi.org/10.1109/IJCNN48605.2020.9207225.

Isabel Wagner and David Eckhoff. Technical privacy metrics: a systematic survey. *ACM Computing Surveys (CSUR)*, 51(3):1–38, 2018. doi: https://doi.org/10.1145/3168389.

Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. Mitigating gender bias in natural language processing: Literature review. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1630–1640, 2019. doi: https://doi.org/10.18653/v1/p19-1159.

Tommi Gröndahl and N Asokan. Text analysis in adversarial settings: Does deception leave a stylistic trace? *ACM Computing Surveys (CSUR)*, 52(3):1–36, 2019. doi: https://doi.org/10.1145/3310331.

Mathias Humbert, Benjamin Trubert, and Kévin Huguenin. A survey on interdependent privacy. *ACM Computing Surveys (CSUR)*, 52(6):1–40, 2019. doi: https://doi.org/10.1145/3360498.

Nektaria Kaloudi and Jingyue Li. The ai-based cyber threat landscape: A survey. *ACM Computing Surveys (CSUR)*, 53(1):1–34, 2020. doi: https://doi.org/10.1145/3372823.

Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3): 1–41, 2020. doi: https://doi.org/10.1145/3374217.

Rada Mihalcea and Carlo Strapparava. The lie detector: Explorations in the automatic recognition of deceptive language. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore, Short Papers*, pages 309–312. The Association for Computer Linguistics, 2009.

Maximin Coavoux, Shashi Narayan, and Shay B. Cohen. Privacy-preserving neural representations of text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1–10, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: https://doi.org/10.18653/v1/d18-1001.

Yangsibo Huang, Zhao Song, Danqi Chen, Kai Li, and Sanjeev Arora. Texthide: Tackling data privacy for language understanding tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1368–1382, 2020. doi: https://doi.org/10.18653/v1/2020.findings-emnlp.123.

Xinghua Zhu, Jianzong Wang, Zhenhou Hong, and Jing Xiao. Empirical studies of institutional federated learning for natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 625–634. Association for Computational Linguistics, 2020. doi: https://doi.org/10.18653/v1/2020.findings-emnlp.55.

Barbara Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004):1–26, 2004.

Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: https://doi.org/10.18653/v1/N19-1423.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. doi: https://doi.org/10.3115/v1/d14-1162.

Armand Joulin, Édouard Grave, Piotr Bojanowski, and Tomáš Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, 2017. doi: https://doi.org/10.18653/v1/e17-2068.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *Advances in neural information processing systems*, pages 5947–5956, 2017.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in neural information processing systems*, pages 3546–3554, 2015.

Qi Feng, Debiao He, Zhe Liu, Huaqun Wang, and Kim-Kwang Raymond Choo. Securenlp: A system for multi-party privacy-preserving natural language processing. *IEEE Transactions on Information Forensics and Security*, 2020. doi: https://doi.org/10.1109/TIFS.2020.2997134.

José Marcio Duarte, Samuel Sousa, Evangelos Milios, and Lilian Berton. Deep analysis of word sense disambiguation via semi-supervised learning and neural word representations. *Information Sciences*, 570:278–297, 2021. doi: https://doi.org/10.1016/j.ins.2021.04.006.

Mozhgan Saeidi, Samuel Bruno da S. Sousa, Evangelos Milios, Norbert Zeh, and Lilian Berton. Categorizing online harassment on twitter. In Peggy Cellier and Kurt Driessens, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 283–297, Cham, 2019. Springer International Publishing. doi: https://doi.org/10.1007/978-3-030-43887-6_22.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: https://doi.org/10.3115/v1/D14-1181.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610, 2005. doi: https://doi.org/10.1016/j.neunet.2005.06.042.

Savelie Cornegruta, Robert Bakewell, Samuel Withey, and Giovanni Montana. Modelling radiological language with bidirectional long short-term memory networks. In *Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis*, pages 17–27, Auxtin, TX, November 2016. Association for Computational Linguistics. doi: https://doi.org/10.18653/v1/W16-6103.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, 2017.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

Jose Camacho-Collados and Mohammad Taher Pilehvar. From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artificial Intelligence Research*, 63:743–788, 2018. doi: https://doi.org/10.1613/jair.1.11259.

Mingqing Chen, Ananda Theertha Suresh, Rajiv Mathews, Adeline Wong, Cyril Allauzen, Françoise Beaufays, and Michael Riley. Federated learning of n-gram language models. pages 121–130, 2019. doi: https://doi.org/10.18653/v1/K19-1012.

Congzheng Song and Ananth Raghunathan. Information leakage in embedding models. *arXiv preprint arXiv:2004.00053*, 2020.

Oren Melamud and Chaitanya Shivade. Towards automatic generation of shareable synthetic clinical notes using neural language models. *NAACL HLT 2019*, page 35, 2019. doi: https://doi.org/10.18653/v1/W19-1905.

Michael D Ekstrand, Rezvan Joshaghani, and Hoda Mehrpouyan. Privacy for all: Ensuring fair and equitable privacy protections. In *Conference on Fairness, Accountability and Transparency*, pages 35–47. PMLR, 2018.

Samuel Sousa, Christian Guetl, and Roman Kern. Privacy in open search: A review of challenges and solutions. In *OSSYM 2021: Third Open Search Symposium*. OSF: The Open Search Foundation, 2021.

Yanai Elazar and Yoav Goldberg. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: https://doi.org/10.18653/v1/d18-1002.

Mihee Lee and Vladimir Pavlovic. Private-shared disentangled multimodal vae for learning of latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1692–1700, 2021.

Eirini Ntoutsi, Pavlos Fafalios, Ujwal Gadiraju, Vasileios Iosifidis, Wolfgang Nejdl, Maria-Esther Vidal, Salvatore Ruggieri, Franco Turini, Symeon Papadopoulos, Emmanouil Krasanakis, et al. Bias in data-driven artificial intelligence systems—an introductory survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(3):e1356, 2020. doi: https://doi.org/10.1002/widm.1356.

Amit Datta, Michael Carl Tschantz, and Anupam Datta. Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. *Proceedings on privacy enhancing technologies*, 2015(1):92–112, 2015.

Orestis Papakyriakopoulos, Simon Hegelich, Juan Carlos Medina Serrano, and Fabienne Marco. Bias in word embeddings. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 446–457, 2020. doi: https://doi.org/10.1145/3351095.3372843.

Masahiro Kaneko and Danushka Bollegala. Gender-preserving debiasing for pre-trained word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1641–1650, 2019. doi: https://doi.org/10.18653/v1/p19-1160.

Luciano Floridi. Establishing the rules for building trustworthy ai. *Nature Machine Intelligence*, 1(6):261–262, 2019.

Junzhe Zhang and Elias Bareinboim. Fairness in decision-making—the causal explanation formula. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. Learning gender-neutral word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4847–4853, 2018.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357, 2016.

Jieyu Zhao, Subhabrata Mukherjee, Kai-Wei Chang, Ahmed Hassan Awadallah, et al. Gender bias in multilingual embeddings and cross-lingual transfer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2896–2907, 2020. doi: https://doi.org/10.18653/v1/2020.acl-main.260.

Joel Escudé Font and Marta R Costa-jussà. Equalizing gender bias in neural machine translation with word embeddings techniques. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 147–154, 2019.

Qiongkai Xu, Lizhen Qu, Chenchen Xu, and Ran Cui. Privacy-aware text rewriting. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 247–257, Tokyo, Japan, October–November 2019. Association for Computational Linguistics. doi: https://doi.org/10.18653/v1/W19-8633.

Yi Chern Tan and L Elisa Celis. Assessing social and intersectional biases in contextualized word representations. In *Advances in Neural Information Processing Systems*, pages 13230–13241, 2019.

Ben Hutchinson, Vinodkumar Prabhakaran, Emily Denton, Kellie Webster, Yu Zhong, and Stephen Denuyl. Social biases in nlp models as barriers for persons with disabilities. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5491–5501, 2020. doi: https://doi.org/10.18653/v1/2020.acl-main.487.

Hila Gonen and Yoav Goldberg. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 609–614, 2019. doi: https://doi.org/10.18653/v1/n19-1061.

C. Basta, M. R. Costa-jussà, and Noe Casas. Extensive study on the underlying gender bias in contextualized word embeddings. *Neural Computing and Applications*, pages 1 – 14, 2020. doi: https://doi.org/10.1007/s00521-020-05211-z.

Chris Sweeney and Maryam Najafian. Reducing sentiment polarity for demographic attributes in word embeddings using adversarial learning. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 359–368, 2020. doi: https://doi.org/10.1145/3351095.3372837.

Chris Sweeney and Maryam Najafian. A transparent framework for evaluating unintended demographic bias in word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1662–1667, 2019. doi: https://doi.org/10.18653/v1/p19-1162.

Malvina Nissim, Rik van Noord, and Rob van der Goot. Fair is better than sensational: Man is to doctor as woman is to doctor. *Computational Linguistics*, 46(2):487–497, June 2020. doi: https://doi.org/10.1162/coli_a_00379.

Oguzhan Gencoglu. Cyberbullying detection with fairness constraints. *IEEE Internet Computing*, 2020. doi: https://doi.org/10.1109/MIC.2020.3032461.

Max Friedrich, Arne Köhn, Gregor Wiedemann, and Chris Biemann. Adversarial learning of privacy-preserving text representations for de-identification of medical records. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5829–5839, Florence, Italy, July 2019. Association for Computational Linguistics. doi: https://doi.org/10.18653/v1/p19-1584.

Maria Barrett, Yova Kementchedjhieva, Yanai Elazar, Desmond Elliott, and Anders Søgaard. Adversarial removal of demographic attributes revisited. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6330–6335, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: https://doi.org/10.18653/v1/D19-1662.

Cornelia Caragea, Ana Uban, and Liviu P Dinu. The myth of double-blind review revisited: Acl vs. emnlp. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2317–2327, 2019. doi: https://doi.org/10.18653/v1/D19-1236.

Ahmadreza Mosallanezhad, Ghazaleh Beigi, and Huan Liu. Deep reinforcement learning-based text anonymization against private-attribute inference. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2360–2369, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: https://doi.org/10.18653/v1/D19-1240.

Oluwaseyi Feyisetan, Tom Diethe, and Thomas Drake. Leveraging hierarchical representations for preserving privacy and utility in text. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 210–219. IEEE, 2019. doi: https://doi.org/10.1109/ICDM.2019.00031.

Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

Stéphane Clinchant, Boris Chidlovskii, and Gabriela Csurka. Transductive adaptation of black box predictions. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 326–331, 2016. doi: https://doi.org/10.18653/v1/p16-2053.

Luca Belli, Sofia Ira Ktena, Alykhan Tejani, Alexandre Lung-Yut-Fon, Frank Portman, Xiao Zhu, Yuanpu Xie, Akshay Gupta, Michael Bronstein, Amra Delić, et al. Privacy-preserving recommender systems challenge on twitter's home timeline. *arXiv preprint arXiv:2004.13715*, 2020.

Elena Battaglia, Livio Bioglio, and Ruggero G Pensa. Towards content sensitivity analysis. In *International Symposium on Intelligent Data Analysis*, pages 67–79. Springer, 2020. doi: https://doi.org/10.1007/978-3-030-44584-3_6.

Fabio Martinelli, Fiammetta Marulli, Francesco Mercaldo, Stefano Marrone, and Antonella Santone. Enhanced privacy and data protection using natural language processing and artificial intelligence. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020. doi: https://doi.org/10.1109/IJCNN48605.2020.9206801.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: evaluating and testing unintended memorization in neural networks. In *Proceedings of the 28th USENIX Conference on Security Symposium*, pages 267–284, 2019.

Xuelong Dai, Hua Dai, Geng Yang, Xun Yi, and Haiping Huang. An efficient and dynamic semantic-aware multikeyword ranked search scheme over encrypted cloud data. *IEEE Access*, 7:142855–142865, 2019. doi: https://doi.org/10.1109/ACCESS.2019.2944476.

Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. Privacy-and utility-preserving textual analysis via calibrated multivariate perturbations. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 178–186, 2020. doi: https://doi.org/10.1145/3336191.3371856.

Hao Liu and Boyang Wang. Mitigating file-injection attacks with natural language processing. In *Proceedings of the Sixth International Workshop on Security and Privacy Analytics*, pages 3–13, 2020. doi: https://doi.org/10.1145/3375708.3380310.

Mayuresh Oak, Anil Behera, Titus Thomas, Cecilia Ovesdotter Alm, Emily Prud'hommeaux, Christopher Homan, and Raymond Ptucha. Generating clinically relevant texts: A case study on life-changing events. In *Proceedings of the Third Workshop on Computational Linguistics and Clinical Psychology*, pages 85–94, 2016. doi: https://doi.org/10.18653/v1/w16-0309.

Yoshua Bengio. *Learning deep architectures for AI*. Now Publishers Inc, 2009. doi: https://doi.org/10.1561/2200000006.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 424–434, 2019. doi: https://doi.org/10.18653/v1/p19-1041.

Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. Disentangling factors of variation in deep representation using adversarial training. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 5040–5048. Curran Associates, Inc., 2016.

Tim Menzies, Ekrem Kocagüneli, Leandro Minku, Fayola Peters, and Burak Turhan. Chapter 16 - how to keep your data private. In Tim Menzies, Ekrem Kocagüneli, Leandro Minku, Fayola Peters, and Burak Turhan, editors, *Sharing Data and Models in Software Engineering*, pages 165–196. Morgan Kaufmann, Boston, 2015. ISBN 978-0-12-417295-1. doi: https://doi.org/10.1016/B978-0-12-417295-1.00016-3.

Lingjuan Lyu, Xuanli He, and Yitong Li. Differentially private representation for nlp: Formal guarantee and an empirical study on privacy and fairness. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2355–2365, 2020. doi: https://doi.org/10.18653/v1/2020.findings-emnlp.213.

Sachin Kumar, Shuly Wintner, Noah A. Smith, and Yulia Tsvetkov. Topics to avoid: Demoting latent confounds in text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4153–4163, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: https://doi.org/10.18653/v1/D19-1425.

Jan Neerbek, Ira Assent, and Peter Dolog. Detecting complex sensitive information via phrase structure in recursive neural networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 373–385. Springer, 2018. doi: https://doi.org/10.1007/978-3-319-93040-4_30.

Natasha Fernandes, Mark Dras, and Annabelle McIver. Generalised differential privacy for text document processing. In *International Conference on Principles of Security and Trust*, pages 123–148. Springer, Cham, 2019. doi: https://doi.org/10.1007/978-3-030-17138-4_6.

Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. Authorship attribution with topic models. *Computational Linguistics*, 40(2):269–310, 2014.

Dainis Boumber, Yifan Zhang, and Arjun Mukherjee. Experiments with convolutional neural networks for multi-label authorship attribution. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

Mohammed Abuhamad, Tamer AbuHmed, Aziz Mohaisen, and DaeHun Nyang. Large-scale and language-oblivious code authorship identification. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 101–114, 2018. doi: https://doi.org/10.1145/3243734.3243738.

Mohammed Abuhamad, Ji-su Rhim, Tamer AbuHmed, Sana Ullah, Sanggil Kang, and DaeHun Nyang. Code authorship identification using convolutional neural networks. *Future Generation Computer Systems*, 95:104–115, 2019. doi: https://doi.org/10.1016/j.future.2018.12.038.

Chandan Akiti, Anna Squicciarini, and Sarah Rajtmajer. A semantics-based approach to disclosure classification in user-generated online content. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online, November 2020. Association for Computational Linguistics. doi: https://doi.org/10.18653/v1/2020.findings-emnlp.312.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Causal mediation analysis for interpreting neural nlp: The case of gender bias. *arXiv preprint arXiv:2004.12265*, 2020.

Accountability Act. Health insurance portability and accountability act of 1996. *Public law*, 104:191, 1996.

Zengjian Liu, Buzhou Tang, Xiaolong Wang, and Qingcai Chen. De-identification of clinical notes via recurrent neural network and conditional random field. *Journal of biomedical informatics*, 75:S34–S42, 2017b. doi: https://doi.org/10.1016/j.jbi.2017.05.023.

Jihad S Obeid, Paul M Heider, Erin R Weeda, Andrew J Matuskowitz, Christine M Carr, Kevin Gagnon, Tami Crawford, and Stephane M Meystre. Impact of de-identification on clinical text classification using traditional and deep learning classifiers. *Studies in health technology and informatics*, 264:283, 2019. doi: https://doi.org/10.3233/SHTI190228.

H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. 2018.

Tao Qi, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie. Privacy-preserving news recommendation model learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1423–1432, Online, November 2020. Association for Computational Linguistics. doi: https://doi.org/10.18653/v1/2020.findings-emnlp.128. URL `https://www.aclweb.org/anthology/2020.findings-emnlp.128`.

Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125. IEEE, 2008. doi: https://doi.org/10.1109/SP.2008.33.

David Cash, Paul Grubbs, Jason Perry, and Thomas Ristenpart. Leakage-abuse attacks against searchable encryption. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 668–679, 2015. doi: https://doi.org/10.1145/2810103.2813700.

Khaled El Emam, Fida Kamal Dankar, Romeo Issa, Elizabeth Jonker, Daniel Amyot, Elise Cogo, Jean-Pierre Corriveau, Mark Walker, Sadrul Chowdhury, Regis Vaillancourt, et al. A globally optimal k-anonymity method for the de-identification of health data. *Journal of the American Medical Informatics Association*, 16(5):670–682, 2009. doi: https://doi.org/10.1197/jamia.M3144.

Elisabeth Eder, Ulrike Krieg-Holz, and Udo Hahn. Code alltag 2.0—a pseudonymized german-language email corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4466–4477, 2020.

Aitor García Pablos, Naiara Pérez, and Montse Cuadros. Sensitive data detection and classification in spanish clinical text: Experiments with bert. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4486–4494, 2020.

Oscar Ferrández, Brett R South, Shuying Shen, F Jeffrey Friedlin, Matthew H Samore, and Stéphane M Meystre. 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, 2013. doi: https://doi.org/10.1136/amiajnl-2012-001020.

Bin Zhou, Jian Pei, and WoShun Luk. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *ACM Sigkdd Explorations Newsletter*, 10(2):12–22, 2008. doi: https://doi.org/10.1145/1540276.1540279.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. doi: https://doi.org/10.1109/TPAMI.2013.50.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, 2004. doi: https://doi.org/10.1145/1014052.1014073.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. In *NIPS Work- shop on Private Multi-Party Machine Learning*, 2016.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.

Brendan McMahan and Daniel Ramage. Federated learning: Collaborative machine learning without centralized training data, 2017. URL `https://ai.googleblog.com/2017/04/federated-learning-collaborative.html`.

Guangneng Hu and Qiang Yang. Privnet: Safeguarding private attributes in transfer learning for recommendation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4506–4516, 2020. doi: https://doi.org/10.18653/v1/2020.findings-emnlp.404.

Ananda Theertha Suresh, Brian Roark, Michael Riley, and Vlad Schogol. Distilling weighted finite automata from arbitrary probabilistic models. In *Proceedings of the 14th International Conference on Finite-State Methods and Natural Language Processing*, pages 87–97, 2019. doi: https://doi.org/10.18653/v1/W19-3112.

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016. doi: https://doi.org/10.1145/2976749.2978318.

Xiao Li, Ye-Yi Wang, and Alex Acero. Learning query intent from regularized click graphs. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 339–346, 2008. doi: https://doi.org/10.1145/1390334.1390393.

Chuan Zhao, Shengnan Zhao, Minghao Zhao, Zhenxiang Chen, Chong-Zhi Gao, Hongwei Li, and Yu-an Tan. Secure multi-party computation: Theory, practice and applications. *Information Sciences*, 476:357–372, 2019. doi: https://doi.org/10.1016/j.ins.2018.10.024.

Ronald Cramer, Ivan Bjerre Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015. doi: https://doi.org/10.1017/CBO9781107337756.

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):9, 2016. doi: https://doi.org/10.1186/s40537-016-0043-6.

Yijia Zhang, Qingyu Chen, Zhihao Yang, Hongfei Lin, and Zhiyong Lu. Biowordvec, improving biomedical word embeddings with subword information and mesh. *Scientific data*, 6(1):1–9, 2019.

Abbas Acar, Hidayet Aksu, A Selcuk Uluagac, and Mauro Conti. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (Csur)*, 51(4):1–35, 2018.

Junyi Li and Heng Huang. Faster secure data mining via distributed homomorphic encryption. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2706–2714, 2020. doi: https://doi.org/10.1145/3394486.3403321.

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, 2014.

Prasha Shrestha, Sebastian Sierra, Fabio A González, Manuel Montes, Paolo Rosso, and Thamar Solorio. Convolutional neural networks for authorship attribution of short texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 669–674, 2017. doi: https://doi.org/10.18653/v1/e17-2106.

Georgios Barlas and Efstathios Stamatatos. Cross-domain authorship attribution using pre-trained language models. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 255–266. Springer, 2020. doi: https://doi.org/10.1007/978-3-030-49161-1_22.

Congzheng Song and Vitaly Shmatikov. Auditing data provenance in text-generation models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 196–206, 2019. doi: https://doi.org/10.1145/3292500.3330885.

Chandler May, Alex Wang, Shikha Bordia, Samuel Bowman, and Rachel Rudinger. On measuring social biases in sentence encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 622–628, 2019. doi: https://doi.org/10.18653/v1/n19-1063.

Ozan Irsoy and Claire Cardie. Deep recursive neural networks for compositionality in language. *Advances in neural information processing systems*, 27:2096–2104, 2014.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018. doi: https://doi.org/10.18653/v1/n18-1202.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf`.

Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Efstathios Stamatatos. A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, 60(3):538–556, 2009. doi: https://doi.org/10.1002/asi.21001.

Steven Burrows, Alexandra L Uitdenbogerd, and Andrew Turpin. Comparing techniques for authorship attribution of source code. *Software: Practice and Experience*, 44(1):1–32, 2014. doi: https://doi.org/10.1002/spe.2146.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537, 2011.

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

Melissa V Harris-Perry. *Sister citizen: Shame, stereotypes, and Black women in America*. Yale University Press, 2011.

Judea Pearl. Direct and indirect effects. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI'01, page 411–420, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558608001.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006. doi: https://doi.org/10.29012/jpc.v7i3.405.

Galen Andrew, Steve Chien, and Nicolas Papernot. Tensorflow privacy, 2019.

Ahmad Al Badawi, Louie Hoang, Chan Fook Mun, Kim Laine, and Khin Mi Mi Aung. Privft: Private and fast text classification with homomorphic encryption. *IEEE Access*, 8:226544–226556, 2020. doi: https://doi.org/10.1109/ACCESS.2020.3045465.