Evaluating the disclosure risk of anonymized documents via a machine learning-based re-identification attack

Benet Manzanares-Salor [*1] [0000-0001-7254-2560], David Sánchez [1] [0000-0001-7275-7887] and Pierre Lison [2] [0000-0002-7649-0689]

[1] Universitat Rovira i Virgili, Department of Computer Engineering and Mathematics, CYBERCAT-Center for Cybersecurity Research of Catalonia Av. Països Catalans 26, 43007 Tarragona, Catalonia, Spain

[2] Norwegian Computing Center, Oslo, Norway

* Corresponding author. E-mail: benet.manzanares@urv.cat
Contributing authors: david.sanchez@urv.cat; plison@nr.no

Abstract

The availability of textual data depicting human-centered features and behaviors is crucial for many data mining and machine learning tasks. However, data containing personal information should be anonymized prior making them available for secondary use. A variety of text anonymization methods have been proposed in the last years, which are standardly evaluated by comparing their outputs with human-based anonymizations. The residual disclosure risk is estimated with the recall metric, which quantifies the proportion of manually annotated re-identifying terms successfully detected by the anonymization algorithm. Nevertheless, recall is not a risk metric, which leads to several drawbacks. First, it requires a unique ground truth, and this does not hold for text anonymization, where several masking choices could be equally valid to prevent re-identification. Second, it relies on human judgements, which are inherently subjective and prone to errors. Finally, the recall metric weights terms uniformly, thereby ignoring the fact that the influence on the disclosure risk of some missed terms may be much larger than of others. To overcome these drawbacks, in this paper we propose a novel method to evaluate the disclosure risk of anonymized texts by means of an automated re-identification attack. We formalize the attack as a multi-class classification task and leverage state-of-the-art neural language models to aggregate the data sources that attackers may use to build the classifier. We illustrate the effectiveness of our method by assessing the disclosure risk of several methods for text anonymization under different attack configurations. Empirical results show substantial privacy risks for most existing anonymization methods.

Keywords: Privacy-preserving data publishing, Re-identification risk, Text anonymization, Language models

Acknowledgments. This research was funded by the European Commission (project H2020-871042 "SoBigData++"), the Norwegian Research Council (CLEANUP project, grant nr. 308904), MCIN/AEI/ 10.13039/501100011033 and "ERDF A way of making Europe" under grant PID2021-123637NB-I00 "CURLING", INCIBE and European Union NextGenerationEU/PRTR (project "HERMES" and INCIBE-URV Cybersecurity Chair) and the Government of Catalonia (ICREA Acadèmia Prize to D. Sánchez).

Author Contributions. Benet Manzanares-Salor: Conceptualization, Methodology, Software, Data Curation, Writing - Original Draft. David Sánchez: Conceptualization, Methodology, Writing - Review & Editing, Funding acquisition. Pierre Lison: Writing - Review & Editing, Funding acquisition.

Conflict of Interests. Authors declare there is no conflict of interests.

1 Introduction

Text is a ubiquitous mean of sharing information among humans, and defines the standard encoding for most unstructured or semi-structured knowledge sources. The availability of text data is therefore crucial for many data mining and machine learning tasks. Examples of textual sources employed for research include medical reports used in pharmacological studies [1], publications in social networks employed for classification [2] or contextual analysis [3], and reviews or comments leveraged for customer satisfaction analysis [4].

Textual data, however, often includes personal information. To comply with the European General Data Protection Regulation (GDPR) [5], privacy protection measures should be undertaken prior to releasing the data or sharing them with third parties. These measures consist on either obtaining the informed consent of the individuals the data refer to (which may be infeasible), or anonymize the data, a process by which it should be not possible to attribute the data to concrete subjects. The latter makes the data no longer personal and, therefore, outside the scope of the GDPR.

Methods for data anonymization have been extensively employed in data mining to conceal personal information in structured datasets consisting of records describing individuals by means of predefined attributes [6-9]. Well-stablished anonymization methods and privacy models for this framework include k-anonymity and its extensions [10-12], and ε-differential privacy [13]. Nonetheless, anonymization of unstructured data such as plain text is significantly more challenging [14, 15]. Difficulties arise because re-identifying attributes appearing in textual resources are unbounded and, usually, not obviously associated to the corresponding subject. Due to these difficulties, text anonymization in real world applications is still mainly performed by human experts, who use their broad language understanding and contextual knowledge to identify and mask the information that may lead to re-identification [16].

The vast majority of automatic text anonymization methods use natural language processing (NLP) techniques to find and mask terms belonging to potentially re-identifying categories [1, 17-31], such as names or addresses. Named entity recognition (NER) techniques are employed to this end. However, because these techniques limit masking to a reduced set of predefined semantic categories -whereas re-identification can be caused by a large variety of entity types-, they offer poor privacy protection. In fact, as introduced above, it is well acknowledged in the data privacy literature that the types of information that may lead to re-identification are unbounded [14, 32]. On the other hand, methods proposed in the area of privacy preserving data publishing (PPDP) [32-38] consider *any* information that jeopardizes subject's anonymity. However, the distortion added by some of these methods and several scalability issues make them unfeasible in many real-world scenarios [14].

In any case, since the majority of text anonymization techniques (and all NLP-based, particularly) do not provide formal privacy guarantees, the level of protection attained should be evaluated *ex post*. In this respect, the *de facto* way to evaluate anonymization methods is to compare them with manual anonymizations [1, 17, 19, 21, 23-26, 28, 30, 33], which assumed to define the ground truth. Particularly, the IR-based metrics *precision* and *recall* are employed for measuring the performance of the anonymization approaches. Whereas a drop in precision indicates that terms were unnecessarily masked (which would negatively affect the utility and readability of the anonymized outcomes), recall, which quantifies the proportion of re-identifying terms that were correctly detected, is inversely related to the re-identification risk. However, recall is not a risk metric, but a completeness measure of a manual anonymization. This renders it severely limited for privacy assessment because *i)* several masking choices -each one involving a different combination of terms- could be equally valid to prevent re-identification, *ii)* recall calculation relies on manual annotations, which are prone to errors, omissions and inconsistencies, and *iii)* not all (missed) terms contribute equally to re-identification [14, 39].

In contrast, the standard evaluation procedure for structured databases in the statistical disclosure control (SDC) field [40] consists of empirically measuring the residual disclosure risk by subjecting the anonymized data to re-identification attacks, more specifically, *record linkage attacks* [41-44]. Record linkage methods, which strictly focus on structured databases, match records in the anonymized database with background data containing publicly available identified information of the protected individuals. Since each successful match between the two data sources results in unequivocal re-identification, the proportion of correctly linked records offers an accurate empirical measurement of the re-identification risk.

Contributions and plan

To tackle the limitations of IR-based metrics and human-dependent evaluation of anonymized texts, in this paper we propose TRIA, a *Text Re-Identification Attack* that adapts the principles of the record linkage technique --which was designed and exclusively applied to structured databases-- to textual documents. Under this framework, we also propose TRIR, a *Text Re-Identification Risk* empirical metric based on the re-identification accuracy of TRIA, which overcomes the drawbacks of the non-risk-specific recall-based privacy evaluation.

We define TRIA as a (highly dimensional) multiclass classification task, where anonymized texts must be associated to individuals (classes) known by the attacker. For maximizing the re-identification capabilities, TRIA leverages state-of-the-art neural language models [45] to aggregate and characterize the data sources that attackers may use as background knowledge to conduct re-identifications. These models have been shown to achieve human or above human proficiency in many language-related tasks, thus making our approach a realistic depiction of an ideal human attacker work. However, because the unconstrained use of large language models can be computationally expensive, we have carefully considered

the means and computational resources expected to be available at the attacker's end in order to simulate a feasible re-identification attack.

We illustrate the effectiveness of our proposal by evaluating the level of privacy protection attained by a variety of text anonymization methods, along with a sample of human-based anonymization [32] under a several (ideal or more realistic) attack configurations. The empirical results reported here illustrate the deficiencies of NER-based anonymization (already discussed at a theoretical level in [14, 32]), and show substantial discrepancies against recall-based evaluations.

The rest of this paper is organized as follows. Section 2 provides background on (text) anonymization and language models. Section 3 discusses related work privacy evaluation for anonymized text. Section 4 presents and formalizes TRIA and TRIR. Section 5 details the implementation details for TRIA. Section 6 reports and discusses empirical results on a variety of text anonymization methods and attack configurations. The last section gathers the conclusions and depicts several lines of future work.

This paper is an extension of the preliminary research reported in a conference paper [46]. The current paper more accurately formalizes the attack and the associated risk metric, and extensively discusses the design and implementation of the leveraged neural model. Moreover, we report a much more comprehensive set of experiments. As a result, Sections 2 and 5 are entirely new. Section 4 has been extended with additional details and a figure. Furthermore, all the experiments and results reported in Section 6 are new.

2 Background

2.1 The anonymization task

Data anonymization is defined as the complete and irreversible removal from a dataset of all the information that could directly or indirectly re-identify the individuals to whom the data refer [47]. Re-identifying attributes can be classified into *identifiers* or *quasi-identifiers*. Identifiers are unique and publicly known values of an individual (such as their complete name or passport number) and, therefore, enable re-identification in isolation. Quasi-identifiers are also publicly known attributes (such as the individual's gender, zip code or birth date) that, even though do not enable re-identification when considered in isolation, may do so when combined with other quasi-identifiers appearing in the same dataset. The quasi-identifiers that can be known on a particular individual constitute the *background knowledge* that can be exploited by attackers to re-identify the subject. Whereas identifying attributes are limited, *any* publicly known information on the individual may act as quasi-identifier and, therefore, quasi-identifying attributes are considered to be unbounded [40].

Anonymization, as it is defined in the GDPR, must therefore remove all identifiers and either remove or mask quasi-identifiers to truly prevent re-identification. Removing just identifying attributes should not qualify as anonymization (because it does effectively prevent re-identification), and goes under the term of *de-identification* [14, 48].

The anonymization task varies significantly depending on the type of data. In structured databases, individuals are described as records, each on containing a fixed (and usually reduced) set of attributes. This structure makes it relatively straightforward for a human expert to classify such attributes into identifiers and quasi-identifiers. Anonymization methods are then applied on the basis of this attribute classification, and their goal is to produce the masking that *best* preserves the utility of the data while enforcing a certain level of privacy protection. Masking consists of suppressing attribute values or replacing them with generalizations.

Database anonymization is usually evaluated in the SDC literature via *record linkage attacks* [41-44, 49, 50]. Record linkage seeks to re-identify protected records by linking the masked quasi-identifying attribute values in the anonymized database with those in the *background knowledge*, which contains the original quasi-identifying values of known individuals. If the linkage is successful, the protected record would be unequivocally associated with the identity of the corresponding individual. Therefore, the re-identification risk of an anonymized database is defined as the proportion of correctly linked records:

$$Re\text{-}identification\ risk = \frac{\#CorrectlyLinkedRecords}{\#Records} \qquad (1)$$

Because quasi-identifying attributes in the anonymized and background databases unequivocally overlap, record linkage can be as straightforward as linking each anonymized record to the record in the background knowledge with the most similar quasi-identifying attribute values [40].

The goal of text anonymization is also to preclude re-identification of the individuals described into the text (*e.g.*, the biographee in a biography text or the patient in a medical report)[1]. However, in unstructured text data, (quasi-)identifiers are not explicitly defined. In fact, almost every word appearing in a text describing the individual to be protected may act as a quasi-identifier [51]. On this basis, the main challenge of unstructured text anonymization is the *detection of text spans* that may act as (quasi-)identifiers (*e.g.*, demographic data on the individuals, spatial-temporal information, etc.), rather

[1] Notice that this differs from the *authorship attribution protection* task, whose goal is to hide the author of a document by employing techniques to mask the author's traits and writing style, which significantly differ from those used in the text anonymization. Works on authorship attribution and its evaluation are therefore outside the scope of this paper.

than the choice of the *masking strategy* [14, 15]. Whereas masking consists on suppressing or replacing the detected (quasi-)identifying text spans by less detailed information (*e.g.*, "New York" → "city", "April 6th" → "April"), the detection of (quasi-)identifiers is performed manually in many practical applications [16]. This makes the latter a costly process that usually involves several human experts, it is prone to errors and omissions and it is often approached as a de-identification -rather than an actual anonymization- task [14, 39].

To alleviate this burden, several automatic methods to detect quasi-identifiers have been proposed. As introduced in the previous section, NLP-oriented methods [1, 17-31] rely on sequence labeling to detect text spans belonging to pre-defined categories that may enable re-identification (typically named entities, such as names or addresses). Detection is based on handcrafted rules or machine learning models trained to identify the occurrence of those specific categories. After that, the detected entities are either removed or masked by replacing them by their categories. Because the types of entities that can be supported by NER models are limited (whereas quasi-identifiers are unbounded), the type of protection offered by these methods is closer to de-identification than to anonymization [14].

On the other hand, PPDP methods operate with an explicit account of disclosure risk and anonymize documents by enforcing a privacy model. As a result, they do not limit the types of information that may act as (quasi-)identifier. However, many of these methods are only applicable in restricted domains [35, 37, 38], suffer from scalability issues due to their dependency on external resources (such as web search engines) [33, 52, 53], or produce distorted word distributions, rather than actual documents, thereby severely hampering the readability and utility of the anonymized outcomes [34, 36]. An exception to the aforementioned methods is [32], which proposes a practical PPDP-oriented approach that does not rely on external resources and retains the structure of the document. For this, it leverages the Word2Vec word embeddings model [54] to efficiently detect text spans that are semantically close to the individual to be protected. More specifically, the model is trained from scratch on the documents to be protected and (optionally) on some public in-domain text. For each training sample derived from a protected document, the corresponding individual name is prepended. This leads to words closely associated with the individual (*i.e.*, identifiers and quasi-identifiers) having embeddings similar to that of the individual's name. On this basis, words in a protected document with a similarity to the individual's name above a predefined threshold are detected as (quasi-)identifiers. These disclosive words are then subjected to masking via ontology-based generalizations.

2.2 Language models

Nowadays, NLP models are rarely built from scratch. Instead, it is common to employ *pre-trained* language models and *fine-tune* them to the specific task to be solved. Language models are statistical models grounded on neural networks that assign probabilities to sequences of words and constitute the state-of-the-art in many NLP tasks, such as translation, text generation, sentiment analysis, summarization, question answering, or text classification [45, 55-57].

Language models are typically based on the Transformer architecture [58], which was introduced as an encoder-decoder architecture for language translation, but has since become the main building block for generic language models. Transformers make it possible to efficiently derive contextualized vector representations (*contextual embeddings*) for each *token* (*i.e.*, word or word fragment) of a given sequence. Those embeddings are computed through multiple neural layers, each being represented by its own set of parameters. Compared to previous neural architectures based on recurrent layers [59, 60], Transformer models can handle long-range dependencies between tokens while simultaneously allowing tokens to be processed in parallel. Consequently, Transformers are better handling longer and more complex texts.

A language model is initially *pre-trained* with a large corpus of documents (*e.g.*, Wikipedia and the BookCorpus [61]) [45, 54], thereby capturing a broad range of text types and linguistic constructions. During pre-training, the model parameters are optimized in an unsupervised fashion from tasks such as masking a randomly selected word and subsequently predicting its value (*masked language modelling*), predicting the next word in a sequence (*causal language modelling*) or determining whether two sentences follow each other in a document (*next sentence prediction*).

Once a general language model is pre-trained, it can be subsequently adapted to a particular task (such as document classification) through a *fine-tuning* process consisting of a supervised learning step on a task-specific corpus. This learning process is a standard neural network training, where the entire training dataset is iterated over multiple times (*epochs*), and multiple training samples (*batches*) are processed in each iteration. Thanks to the general linguistic and factual knowledge acquired during pre-training, satisfactory results can be achieved even if the amount of labeled data available for fine-tuning is small. In addition, results can be improved if, previously to fine-tuning, the language model is additionally pre-trained using the task data. This process, which is conducted with the same training method employed to general pre-training, adapts the model to the specific task domain.

In recent years, the HuggingFace's Transformers library [62] has emerged as the standard free platform for the implementation, pre-training, fine-tuning, usage and sharing of (pre-trained) language models. One of the most popular and well-established transformer-based language models available are BERT (*Bidirectional Encoder Representations from Transformers*) [45] and its variations, which either improve its performance [63] or reduce the model's size and, therefore, its computational requirements [64]. After fine-tuning, BERT and its variations have demonstrated human-level performance and competitiveness with larger language models [65] in multiple language-related tasks, including text classification. All these language models are freely available at Hugging Face's transformers library [62] with pre-trained weights and an easy-to-use implementation.

3 Related work

The vast majority of text anonymization methods proposed in the literature do not offer formal privacy guarantees [39]. Consequently, the level of privacy protection attained should be evaluated *ex post*, that is, through empirical evaluations of the re-identification risk of the protected outcomes. The objective is to determine whether the anonymization meets the privacy requirements set by the data holder.

Because IR-based metrics such as precision and recall constitute the standard approach to evaluate many sequence labelling tasks (and NER in particular), and NER models are the most common methods employed to tackle text anonymization, works in the literature rely on the recall metric to evaluate the attained privacy [1, 17, 19, 21, 23-26, 28, 30, 33]. *Recall* is an IR-based completeness metric defined as the proportion of relevant elements that were correctly detected by the system:

$$Recall = \frac{\#TruePositives}{\#TruePositives + \#FalseNegatives} \quad (2)$$

where *#TruePositives* is the number of relevant elements detected by the system and *#FalseNegatives* is the number of missed ones with respect to a ground truth. In text anonymization, these relevant elements are the (quasi-)identifying text spans that should be detected (and subsequently masked), and the ground truth consists of manual annotations of (quasi-)identifiers performed by human experts.

However, employing recall for privacy evaluation has several drawbacks [39, 66]. Specifically, recall does not assess the residual re-identification risk of anonymized texts, but only compares the anonymized outcomes against manual annotations. These annotations are subjective, and may be prone to errors and omissions [14, 39]. Moreover, manual annotation is time consuming, and necessarily involves several human experts, whose annotations should be integrated through a non-trivial and non-univocal process [39]. Another drawback of IR-based metrics is that they assume the existence of a single gold standard annotation. Even though this assumption may be reasonable for NER or other NLP-oriented tasks, it does not hold for text anonymization, where several masking choices (each one encompassing a different combination of quasi-identifying terms) could be equally valid to prevent re-identification [14]. Finally, recall-based evaluation assumes that all identified/missed elements contribute equally to mitigate/increase re-identification risk, which is rarely the case [39]. Indeed, failing to mask an identifier (such as the full name of an individual) is much more disclosive than just missing the birthdate or city name.

4 Re-identification attack and disclosure risk metric for anonymized text

In the following we present TRIA, a *Text Re-Identification Attack* for (anonymized) texts based on large neural language models, and TRIR, a *Text Re-Identification Risk* metric based on TRIA's accuracy. Both aim to provide a practical solution to evaluate the actual privacy protection offered by anonymization methods for textual data. Compared to recall, our approach brings the following advantageous features:

- *Self-supervision*: TRIR does not require from costly manual annotations as anonymization ground truth, which can be imperfect, potentially biased (if coming from a single annotator) or inconsistent (if coming from multiple annotators). Once the protected and background documents are set, our metric is fully automatic. Supervised learning only requires the label indicating the individual to be protected for each document.
- *Focus on risk*: Contrary to recall, our metric directly measures the empirical re-identification risk of (protected) documents as a whole, rather than assessing the coverage of individual annotations.
- *Flexibility*: Our metric can be tailored to take into account different resources available to attackers (see Section 4.2) thereby allowing to configure either ideal or more realistic attackers.

TRIA tries to re-identify the subjects referred in a collection of anonymized texts by leveraging a multiclass classifier, with one class for each individual known by the attacker. A set of identified and publicly available texts is employed for training the classifier. These texts encompass a population of known subjects in which the individuals referred in the anonymized set are expected to belong to. For instance, to re-identify anonymized medical reports from a city hospital, the attacker may use publicly available social networks posts from the residents of that city. The set of known individuals should therefore be a superset of the anonymized subjects. The anonymized documents would contain masked quasi-identifiers (*e.g.*, age intervals) and confidential attributes (*e.g.*, diagnoses) from unidentified individuals, whereas the public texts would contain identifiers (*e.g.*, names and surnames) and clear quasi-identifiers (*e.g.*, specific ages) from known individuals. Deficiencies in anonymization would enable unequivocal matchings of the (quasi-)identifiers of both types of documents, allowing to re-identify the subjects in the protected documents and, subsequently, disclose their confidential information.

According to the above, our approach can be seen as an adaptation of the record linkage attack (designed for structured databases) to textual data, where words or text spans correspond to attribute values, documents correspond to records, and the classifier performs the linkage between the anonymized and public data.

4.1 Formalization

Let A_D be the set of anonymized (*i.e.*, non-identified) documents and B_D the set of identified public documents (*i.e.*, background knowledge). Each document refers to a specific subject, thereby defining the sets of individuals A_I and B_I, and the mapping functions F_A: $A_D \rightarrow A_I$ and F_B: $B_D \rightarrow B_I$.

As in the original record linkage attack, we assume that $A_I \subseteq B_I$, which means that a re-identification function F_C: $A_D \rightarrow B_I$ exists, which matches protected documents with the corresponding known individuals. From the attacker's point of view, A_D, B_D, B_I and F_B are known, and A_I, F_A and F_C are unknown. Consequently, the goal of the attack is to exploit the similarities between A_D and B_D to obtain F_C', an approximation of F_C. Figure 1 depicts a graphical representation of this setting.

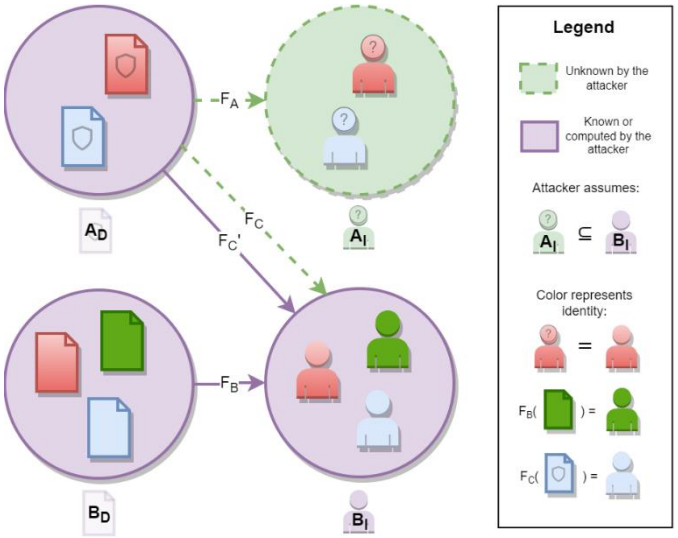


Figure 1: TRIA scenario, including individuals and documents sets from the point of view of the attacker

Using a medical example above, the anonymized documents (A_D) might be patient medical reports, and the individuals to be protected (A_I) would be those patients. The identified documents (B_D) are publicly available documents associated with the individuals' identities, such as posts of such patients about themselves in their social network accounts. The individuals linked to the public documents (B_I) are known to the attacker (social networks are non-anonymous and publicly accessible), while the patients (A_I) and their connection to specific medical reports --which correspond to F_C, the re-identification function--, are initially unknown.

Our proposal is formalized in Algorithm 1, which outputs the number of correct re-identifications achieved by TRIA for a collection of anonymized documents.

Algorithm 1 Re-identification risk assessment algorithm for anonymized text documents

Require: A_D, B_D, B_I, F_B, F_C
1: $classf \leftarrow build_classifier(B_D, B_I, F_B)$ /* Training of TRIA */
2: $numReIds \leftarrow 0$ /* Number of correct re-identifications */
3: **for each** d in A_D **do**
4: $pred_ind \leftarrow classf.predict(d)$ /* Predicted individual by TRIA */
5: **if** $pred_ind == F_C(d)$ **then**
6: $numReIds \leftarrow numReIds + 1$
7: **end if**
8: **end for**
9: **return** $numReIds$

First, a multiclass classifier is built and trained to predict F_C (line 1, see Section 5.4 for details). Following the formal notation above, the classifier would implement F_C' by learning which individuals from B_I correspond to the documents in B_D based on the knowledge (*i.e.*, data sources) available to the attacker. Afterwards, the classifier is tested on the anonymized set of documents A_D (line 4, refer to Section 5.5 for details). A correct re-identification would happen if the prediction (*i.e.*, F_C') matches F_C (lines 5-6). The total number of re-identifications is finally returned in line 9.

As in the record linkage method (Eq. 1), we assess the TRIR of A_D as the accuracy of TRIA:

$$TRIR = \frac{numReIds}{|A_I|} \qquad (3)$$

where *numReIds* is the number of correct re-identifications achieved by Algorithm 1, and $|A_I|$ is the number of protected individuals from the A_I set.

4.2 Attacker's resources

One of our goals for TRIA is to be flexible enough to encompass both ideal or more realistic attackers. This means considering the resources (*i.e.*, background knowledge and available computational resources) that attackers may (ideally or realistically) devote to the attack. This is in line with the GDPR[2], which states that, to assess the residual risk of anonymized data, one should account the reasonable means that can be employed to achieve re-identifications.

We next characterize and discuss the resources that attackers may use to conduct re-identification attacks, and that should be considered when building TRIA's classifier:

- *Background knowledge breadth*: Corresponds to the number and type of individuals (and corresponding documents) considered as candidates for re-identification or, in other words, to the size of the population to which the individuals to be protected belong. Formally, this is the B_I set (and corresponding B_D) that the attacker determines as superset of A_I. It is important to note that the $A_I \subseteq B_I$ assumption may not hold in practice because A_I would be usually unknown to the attacker. In this case, the re-identification risk is limited by the proportion of overlap between A_I and B_I. Furthermore, the difficulty of the re-identification task would depend on the number of individuals and the intra-class similarity of both A_I and B_I. On the one hand, increasing the size of B_I would complicate the re-identification, since there are more individuals (classes) to distinguish. This behavior is inherent to any multi-class classification task. In addition, the more individuals in B_I not present in A_I the more prone to false positives the classifier would be; this increases the cost of the attack (because more classes must be learned) with no expected benefits. On this basis, the goal of an attacker would be to define a B_I set with a size as close as possible to that of A_I while trying to fulfill $A_I \subseteq B_I$. On the other hand, a high intra-similarity between the individuals in B_I not present in A_I does not affect the accuracy of the attack since there are no instances for these individuals in A_D. Finally, regarding the intra-similarity between the individuals in B_I also present in A_I, and those not present in A_I, the lower the better, since we do not want the classifier to get confused by individuals outside A_I.
- *Background knowledge depth*: Encompasses the quantity and quality of the background information available to attackers for each individual or, in other words, the (quasi-)identifiers available in each document of B_D. Generally, the more information the attacker has for an individual, the easier the re-identification will be. However, if the individual in B_I is not represented in A_I (*i.e.*, a background knowledge breadth issue), the information will probably render irrelevant and misleading for the classifier.
- *Computational capabilities*: Processing power available to attackers influences their ability to perform inferences. Greater computational capacity would enable them using more complex models to build the classifier and/or more costly training strategies (see next section), potentially increasing the success rate of the attack.

In the next section we discuss the choices that attackers may make to balance inference accuracy and computational cost. Moreover, the experiments we report in Section 6 are defined so that the influence of these resources in the TRIA's accuracy can be observed in practice.

5 Choosing and implementing the classifier model

The classifier model is a fundamental element of our approach because it determines the accuracy of the attack according to the resources available to execute it. Even though the design of TRIA is general and may be enforced by means of any machine learning model capable of classifying documents, in this section we present our specific design choices and implementation details adopted for the *build_classifier* and *predict* methods (lines 1 and 4 of Algorithm 1).

5.1 Selecting the language model

Our goal is to reproduce as realistically as possible the approach that an attacker may employ for re-identification. Under this premise, we expect attackers to take advantage of pre-trained language models, which have been introduced in Section 2.2, as state-of-the-art for NLP. The idea is to exploit the general understanding of text brought by language models devise a re-identification attack by additionally pre-training and fine-tuning those models with the background knowledge

[2] Extract from the GDPR's Recital 26: "[…] To determine whether a natural person is identifiable, account should be taken of all the means reasonably likely to be used, such as singling out, either by the controller or by another person to identify the natural person directly or indirectly. To ascertain whether means are reasonably likely to be used to identify the natural person, account should be taken of all objective factors, such as the costs of and the amount of time required for identification, taking into consideration the available technology at the time of the processing and technological developments. […]"

available.

On this basis, we consider BERT [45] or its variations [63, 64, 67-69] as reasonable language models candidates for the attack. Table 1 compares widely used language models grounded on BERT in terms of size and pre-training data, which we will later use in our empirical experiments. In particular, DistilBERT [64] is a distilled version of the original BERT which reduces a 40% the model's size while keeping a 97% of its performance in multiple tasks, while RoBERTa [63] is a replication study of BERT pre-training, with a careful search of hyperparameters and trained with ten times more data (including also BERT's training set). Due to its larger size, RoBERTa tends to provide better results than BERT in multiple tasks. BERT-based models can obtain human-level results with neither a huge cost nor unfeasible knowledge assumptions from the attacker. They can also be easily run locally, thereby eliminating the need to outsource potentially sensitive documents to a remote server for processing. Furthermore, these models are well-established and publicly accessible, with numerous domain-specific fine-tuned versions available[3].

Table 1: Comparison of widely-used BERT-based language models

Name	#parameters	Pre-training data
BERT [45]	110M (bert-base-uncased) 340M (bert-large-uncased)	Wikipedia + BookCorpus [61]
DistilBERT [64]	66M (distilbert-base-uncased)	Same as BERT
RoBERTa [63]	125M (roberta-base) 355M (roberta-large)	Same as BERT + CCNews [70]+ OpenWebText [71] + Stories [72]

The following subsections explain our approach for using a BERT-based language model for re-identification. This approach can be applied to any BERT-based language model with negligible code modifications thanks to the HuggingFace's Transformers library [62] we employ, which is the de facto standard library for the implementation, pre-training, fine-tuning and usage of language models.

5.2 From language model to classifier model

To adapt a BERT-based language model for a classification task, we apply the standard procedure described in [45] already implemented in the Transformers library[4]. It consists of appending a fully connected classification layer to the language model. This layer takes as input the output vector corresponding to a special classification token named [CLS] that is prepended to the input sequence [45]. The classification layer then produces a logits vector with one value per class (which, in our case, corresponds to an individual from B_I). By applying the normalization SoftMax activation to the logits, the final probability for each class is obtained. As this new layer is designed to identify the individual from B_I related to the document, we will refer to it as the *individual classification layer*.

Transformer-based models such as BERT have a limited input length. In the case of BERT (and its variations) this limit is 512 tokens, which is shorter than the length of many real-world documents. To address this limitation, we divide each document into acceptable-length splits and process each split using the BERT-based classifier. Details of the splitting criterion are provided in the next section. During training, the BERT-based classifier is fine-tuned to classify document splits individually (see Section 5.4). For prediction, we aggregate the classification output of each split to obtain the final prediction. More details on this are provided in Section 5.5.

5.3 From documents to model input

For each input document we perform the three subsequent steps:

- *Pre-processing*: This step is done to unify semantically equivalent text spans and remove non-meaningful ones. Concretely, we perform lemmatization (*e.g.*, "changing" replaced by "change") and remove special characters (*e.g.*, the "#" of a hashtag) and stop words (*e.g.*, "the" or "that"). This can be done with any NLP library such as spaCy[5]. On the one hand, lemmatization gets rid of morphological variations of words, making the classification less dependent on the text syntax and more focused on the ((quasi-)identifying) facts conveyed by the document. On the other hand, the removal of special characters and stop words brings three benefits. First, it discards non-meaningful (and, therefore, non-identifying) information that may confuse the model. Second, since the classifiers' input length is limited, removing those elements increases the density of quasi-identifiers at each split/input, which should be beneficial for re-identification. Finally, it reduces the number of splits (explained at the sliding window step) generated per document that, in turn, decreases the training and prediction runtimes.

[3] https://huggingface.co/models?other=bert
[4] https://huggingface.co/docs/transformers/v4.40.1/en/model_doc/bert#transformers.BertForSequenceClassification
[5] https://spacy.io/

- *Tokenization*: Transforms the sequence of words of the input text into a sequence of tokens, as required for a language model. For BERT-based models, the standard WordPiece subword segmentation algorithm (which BERT relies upon [45]) is employed by leveraging the implementation available in the Transformers' library[6].
- *Sliding window*: To accommodate BERT-based models' input length limitations (512 tokens), tokenized documents sometimes need to be divided into smaller splits. Inspired by [73], we split documents by using a sliding window with a length of *N* tokens (being *N* smaller or equal than the model's limit), and an overlap of *M* tokens with the previous window (equivalent to a stride of *N-M*). The overlap enables the model to never lose the context of consecutive tokens, that is, it ensures that the model can observe the previous and following *M* tokens for every token (except the document's first and last) in at least one split. Setting an appropriate *N* and *M* is crucial, since they determine how long the detected long-term dependencies can be (which affects the model's accuracy), and the number of text splits generated (that influences training and prediction runtimes, which depend on the number of inputs rather than their length). These *N* and *M* hyperparameters can be empirically adjusted through hyperparameters search, as we detail in Section 5.6.

5.4 Building the classifier

By leveraging the steps outlined in the previous subsections, we now detail the implementation of the *build_classifier* method from Algorithm 1. As shown in the algorithm, it receives as input the background knowledge documents (B_D), the corresponding individuals (B_I) and the mapping function (F_B) that connects both. The output of the method is a classifier trained for document re-identification. In Figure 2, a diagram of the workflow for this method is depicted, which we explain below.

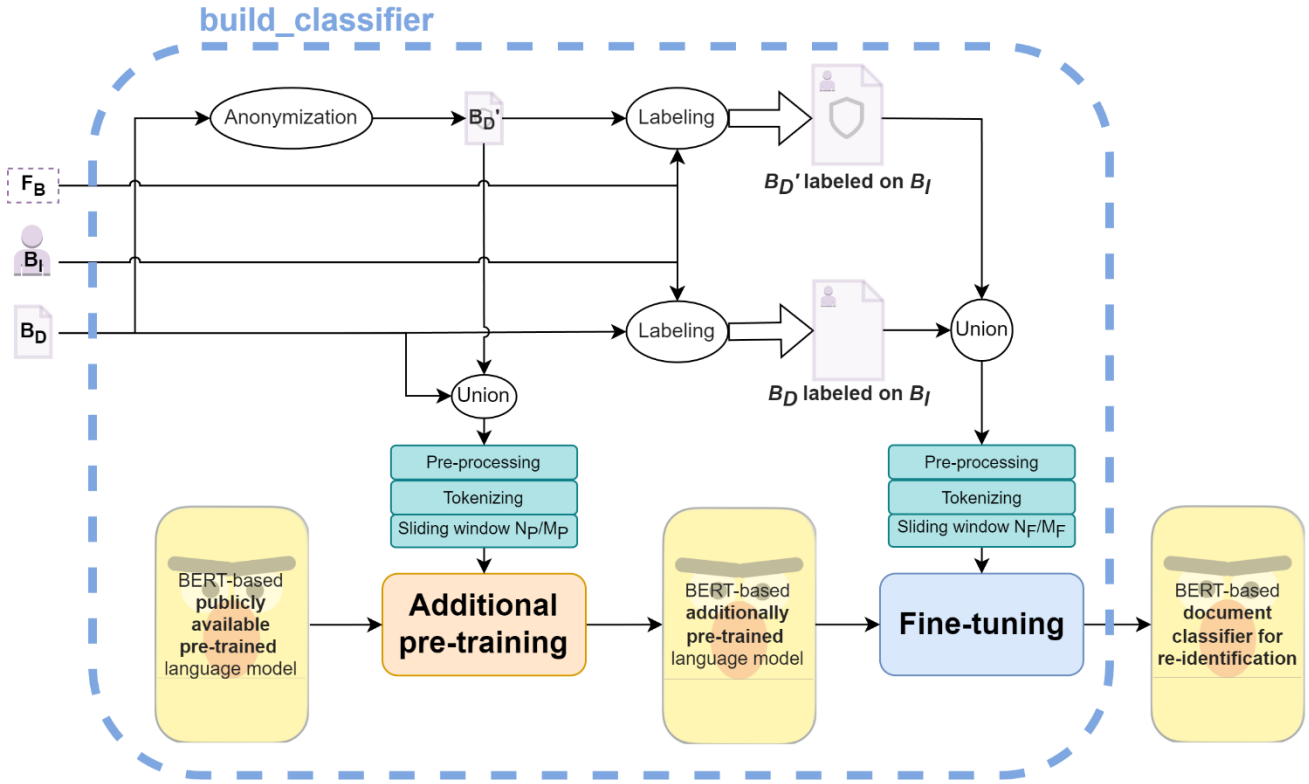


Figure 2: Workflow for the *build_classifier* method from Algorithm 1

As introduced in Section 2.2, adapting language models to specific tasks often involves additional pre-training and/or fine-tuning steps on task data. Following [74], we apply both steps to the language model for obtaining a corpus-specific BERT classifier adapted to the domain of the documents to re-identify. For both steps, the procedures defined in Section 5.3 are applied to all documents identically with the exception of sliding window splitting.

In the additional pre-training step, the model is further optimized on a masked language modelling (MLM) objective. This the standard pre-training task in which BERT was initially trained on, and implementations[7] and guides[8] are provided in the Transformer's library. In MLM, a vocabulary classification layer is appended to the language model, and all the

[6] https://huggingface.co/docs/transformers/model_doc/bert#transformers.BertTokenizerFast
[7] https://huggingface.co/docs/transformers/v4.40.1/en/model_doc/bert#transformers.BertForMaskedLM
[8] https://huggingface.co/docs/transformers/tasks/masked_language_modeling

parameters of this extended model are optimized. At the end, the vocabulary classification layer is removed, and we preserve the additionally pre-trained language model. For this step, increasing the sliding window length and/or overlap (N_P and M_P in Figure 2) should be beneficial, so that the context available for each token is maximized.

For the fine-tuning step, the language model resulting from the additional pre-training step is further optimized on labelled data[9], which in our case corresponds to the identity of the individuals referred in the background documents. More specifically, an individual classification layer is appended to the language model (see Section 5.2), and the resulting classifier model is trained with a cross-entropy loss[10] to predict the individual being referred to in each document split. For this step, increasing the length of the sliding window (N_F in Figure 2) may not be beneficial: whereas a longer window provides more information to assist the classification, it can also be too specific to the document. Moreover, for a fixed overlap (M_F in Figure 2) ratio, the longer the window, the fewer splits are created for each document, thus reducing the variability of the samples per individual. As a result, a too large window may lead to overfitting, and make the model focus on document-specific (sub)sentences rather than on more general (quasi-)identifying words or text spans. The same sliding window configuration selected for fine-tuning will be used at prediction (see Section 5.5).

Regarding the data employed for this two-step training, it should be stressed that the attackers' knowledge is limited to B_D, B_I and F_B. Documents in B_D provide knowledge of the individuals' specific vocabulary, which improves understanding of domain-specific words. Additionally, B_D can be labeled on B_I by using F_B, thereby providing useful information about the relationship between the publicly available information and the individuals' identity. This can lead to the detection of (quasi-)identifying attributes (*e.g.*, the person's name or demographic attributes), which form the basis of the re-identification attack.

An intuitive approach would be to perform additional pre-training using B_D and fine-tuning with B_D labeled on B_I. This results in a corpus-specific model capable of mapping documents to B_I, as it is needed for the attack. However, it is important to note that the goal of the attack is to correctly classify documents from A_D, which stem from a different data distribution than B_D documents. In particular, B_D are clear texts (*e.g.*, identified publications in social networks) whereas A_D are anonymized documents (*e.g.*, non-identified medical reports with quasi-identifying terms masked via suppression or generalization). This could impair the re-identification accuracy, since machine learning algorithms are sensitive to differences between training and test data distributions.

Our solution is to create B_D', an anonymized version of B_D, by using any practical text anonymizer that attackers may have at their reach. Using the same method employed for A_D would be the ideal but, since such method would be rarely known, a standard NER-based approach (given that NER is the most used and directly applicable technique for text anonymization) may be employed instead. In our experiments, we utilized spaCy NER for this purpose, which is one of the evaluated anonymization techniques (refer to Section 6.1). Because the resulting document set B_D' is more similar to A_D, it will provide a better approximation of how data are (typically) anonymized. Moreover, B_D' can be labeled on B_I (given that $B_D' \rightarrow B_D$ is known), which facilitates discovering the identities associated to the masked documents; for example, by discovering identifying words neglected by the anonymization method (*e.g.*, a specific location) that also appear in A_D documents. Under these considerations, we propose using B_D and B_D' for additional pre-training, and the union of B_D and B_D' labeled on B_I for fine-tuning, thereby obtaining a classifier model better adapted to the expected anonymized documents.

5.5 Prediction

The classifier resulting from the *build_classifer* method is employed in the *predict* method of Algorithm 1 to infer the individual from B_I corresponding a (presumably anonymized) document. This final classification step is depicted in Figure 3. First, the document is pre-processed, tokenized and split by using the same sliding window configuration employed for fine-tuning (see Section 5.4). Afterwards, each split is processed by the trained BERT-based classifier (one at a time) and the results are aggregated into a single global prediction using the element-wise sum of all the logits vectors. The distribution of class (in our case equivalent to individuals in B_I) probabilities is obtained by applying a SoftMax function to those results. Finally, the class/individual with the maximum probability is chosen as the predicted individual for the document.

We favored the sum of logits instead of the most frequent prediction or the sum of the classes' probabilities (*i.e.*, logits after SoftMax) to account for the magnitude of the output. If the model is able to clearly identify an individual based on one text split, this prominent output would be reflected in the sum of logits. The motivation for this design is to mimic human reasoning for re-identification, which gives greater importance to parts of the text with clear re-identification clues (such as (part of) the person's name left unmasked), than to those with less clear evidences. This is also in line with [10, 66], which point out that missing a single identifying attribute during anonymization may be enough to reveal an individual's identity.

[9] https://huggingface.co/docs/transformers/es/training
[10] https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html

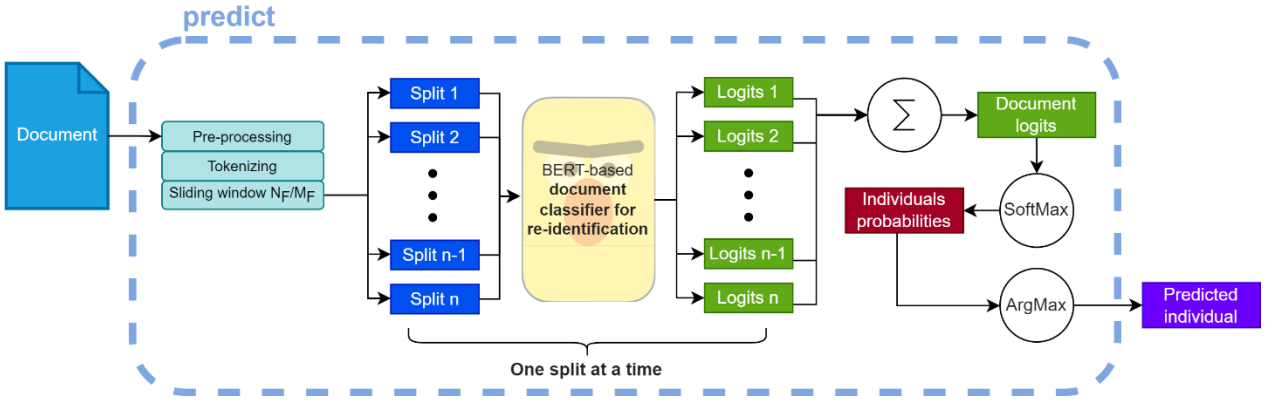


Figure 3: Our proposed workflow for the predict method

5.6 Selection of hyperparameters

The classifier construction relies upon the definition of several hyperparameters common to neural networks training (*i.e.*, number of epochs, *learning rate* and *batch size*, see Section 2.2) and TRIA-specific (*i.e.*, *sliding window length* and *sliding window overlap*, see Section 5.2). To this end, we propose to empirically set this hyperparameters based on a separate development set, as it is standard in machine learning. That is, by defining a development set disjoint but similar to the evaluation set and performing multiple tries with different hyperparameter values in order to select the combination that provided the best results.

In our case, the evaluation set to be mimicked by the development set are the documents in A_D labeled in B_I (*i.e.*, re-identification ground truth) and a try is the complete re-identification attack presented in Algorithm 1. We propose to define a fixed number of epochs for additional pre-training, and fine-tune it until a pre-defined maximum number of epochs is achieved or the development accuracy stops improving (early stopping). As development set, we propose to create a random subset from the documents in B_D, which we call C_D, and transform it to be as similar as possible as the data distribution of A_D. For this, a straightforward approach would be to anonymize C_D; nevertheless, this would result into identical texts to those in B_D', which are already present in training data. Thereupon, a previous step is required aiming to differentiate C_D texts from B_D ones and, if possible, to assimilate them to those in A_D prior anonymization. We propose to apply a summarization-like procedure on documents from C_D, which would result in a set that we call $Ç_D$. To this end, abstractive or hybrid summarization methods are preferred to extractive ones [75], so the resulting summarizations do not include sentences identical to those present in B_D. After summarization, documents in $Ç_D$ are anonymized (producing $Ç_D$') as done for B_D'. The resulting $Ç_D$' is used as the development set. The specific hyperparameters employed in our experiments are detailed in Section 6.3.

6 Evaluation results

In the following we report empirical results on using TRIA and TRIR to evaluate several text anonymization methods on a common corpus of text documents. To offer a broad comparison, we assessed the residual re-identification risk of NLP-based, PPDP-oriented and manual anonymizations. Our experiments are configured to provide a comprehensive analysis of the influence of the resources available to attacker resources in the risk assessment. We also report a comparison against the standard recall metric based on ground truth human annotations. To ensure reproducibility, all the code and data are available on: https://github.com/BenetManzanaresSalor/TextRe-Identification

6.1 Evaluated methods

As stated in Section 2.1, NLP-based methods [1, 17-31] usually approach anonymization as a NER task, in which words belonging to allegedly re-identifying categories (*e.g.*, locations, names, dates, etc.) are masked. In this case, masking consists of replacing the detected entities by their semantic categories. We evaluated the following tools for NER-based text anonymization [14]:

- *Stanford NER* [76]: offers three pre-trained NER models: *NER3*, which is able to detect LOCATION, ORGANIZATION and PERSON types; *NER4*, which detects LOCATION, ORGANIZATION, PERSON and MISC (miscellaneous); and *NER7*, which detects ORGANIZATION, MONEY, DATE, PERCENT, PERSON and TIME.
- *Microsoft Presidio*[11]: an anonymization-specific tool based on NER. Among the diversity of entity types supported by Presidio, we enabled those corresponding to quasi-identifying information: PERSON, LOCATION, NRP -person's nationality, religious or political group- and DATE_TIME.

[11] https://github.com/microsoft/presidio

- *spaCy NER*[12]: we employed the *en_core_web_lg*[13], neural model trained on the Ontonotes v5 corpus [77], which is capable of detecting named entities of DATE, CARDINAL, EVENT (*e.g.*, wars), GPE (*e.g.*, countries, cities), FAC (*e.g.*, buildings), LANGUAGE, LAW (named documents made into laws), MONEY, LOC (non-GPE locations such as mountain ranges), NORP (nationalities or religious or political group), ORDINAL, PERCENT, ORG, PERSON, PRODUCT (*e.g.*, vehicles), TIME (times smaller than a day), QUANTITY and WORK_OF_ART (*e.g.*, books titles, songs) types.

For PPDP-grounded text anonymization methods, as discussed in Section 2.1, the only practical method we found is [32], which is based on word embedding models. Hereafter, we will refer to this method as Word2Vec, as it is built upon this neural model. In comparison with the fixed and non-configurable anonymization of NER-based methods, Word2Vec stands out as the only method evaluated that can be tuned to adjust the trade-off between privacy and utility. This is controlled by a threshold t that determines the minimum embedding similarity required for a word to be masked. In our experiments, we employed two threshold values: t=0.25, which was observed by the method's creators to maximize privacy protection, and a more relaxed t=0.5, which would result in less masking and, therefore, weaker protection. For this method, we leveraged its implementation available in GitHub[14].

In addition to the methods listed above, we also considered the manual anonymization effort conducted in [32], which was based on sound privacy-oriented annotation guidelines. This enables us to evaluate the robustness of manual anonymization under TRIA, but also makes it possible to compute the recall of the above-described methods -by considering the manual anonymization as the ground truth-, and compare this metric against the risk assessment of TRIR.

Finally, we also report the TRIR for the non-anonymized versions of the documents in A_D. This defines the baseline risk that anonymization methods should (substantially) reduce.

6.2 Evaluation corpus and background documents

The corpus described in [32] was employed as evaluation data. It consists of 18,672 Wikipedia articles corresponding to the "20th century actors" Wikipedia category. Each one provides biographical details of a movie actor. To simulate the scenario depicted in Figure 1, we considered the article abstracts as the texts to be anonymized, and the article bodies, which provide extended details on the abstract's information, as the identified publicly available documents on the individuals to be protected. A subset of 50 article abstracts corresponding to 50 contemporary, popular and English speaking actors was selected from this corpus (as done in [32]). We considered them the set of documents to be anonymized. In attack's formal notation, the 50 extracted actors constitute A_I, the 50 abstracts anonymized with a specific method m define A_D^m, and the article bodies in the corpus define B_D with a population of B_I actors.

As introduced in Section 4.2, TRIA (and subsequently TRIR) depends, among other attacker's resources, on the *background knowledge breadth*; that is, the overlap between B_I (the complete set of known individuals in the background knowledge) and A_I (the set of individuals referred to in the anonymized documents, which are unknown to the attacker), and the relative size and similarity of those two sets. To evaluate the background knowledge breadth, we defined several scenarios by setting increasingly larger B_Ds and corresponding B_Is with varying distributions:

- *50_eval*: this is a worst-case scenario for privacy, in which B_I exactly matches A_I. Subsequently, B_D comprises the 50 article bodies from Wikipedia corresponding to the 50 anonymized abstracts from those same articles. This scenario constitutes the easiest re-identification setting because there are no individuals in B_I absent from A_I and, therefore, defines an ideal attacker with respect to background breadth.
- *500_random*: a scenario consisting of 500 article bodies randomly extracted from the full corpus plus those corresponding to the 50 actors in A_I that were not included in the random selection (thereby ensuring that $A_I \subseteq B_I$). Compared to the *50_eval* scenario, B_I is larger, with most of its individuals absent from A_I. Nevertheless, the similarity between the *randomly selected* individuals not present in A_I and the *popular English-speaking* actors of A_I is expected to be low, thereby facilitating re-identification. This scenario defines a more feasible but still powerful attack.
- *500_filtered*: a more realistic scenario with 581 article bodies obtained by filtering the full dataset of 18,672 articles according to several attack-oriented criteria. The criteria aim at maximizing the number of individuals in A_I present in B_I (even without knowing A_I, as it would happen in practice). In particular, the filtering discarded article bodies corresponding to non-native English speakers, dead individuals, non-actors (*e.g.*, directors or producers), actors born before 1950 or after 1995 (latter included) and those whose article included less than 100 links and was available in less than 40 languages. The latter two criteria aim to capture the 'popularity' of the actor. As a result, only 40 out of the 50 actors in A_I appeared in B_I, thereby limiting re-identification accuracy to 80%. Higher intra-similarity than for the *500_random* scenario is expected due to the tailored (*i.e.*, non-random) filtering. This defines a more realistic scenario, because the attacker knows that A_I individuals are popular,

[12] https://spacy.io/api/entityrecognizer
[13] https://spacy.io/models/en#en_core_web_lg
[14] https://github.com/fadiabdulf/automatic_text_anonymization

English-speaking and alive actors. The attack is considered a powerful yet credible threat.

- *2000_filtered*: a more realistic scenario with 1,952 article bodies selected by using the same filtering as in *500_filtered* but omitting the criterion related to the number of languages per article. As a result, 41 out of the 50 actors in A_I appeared in B_I, thereby limiting re-identification accuracy to 82%. This scenario is a superset of *500_filtered*, with 1,371 new individuals added due to the less strict filtering, most of them not present in A_I. The intra-similarity is expected to be slightly lower than between the 581 individuals of *500_filtered*, but still significantly higher than for *500_random*. This defines a weaker but probably more feasible attack, with a more realistic size for the population in the background knowledge.

As we latter report, for larger cardinalities of B_D the training time of the model is likely to take longer than 12 hours with the (reasonably powerful) hardware configuration we employed. Specifically, ten fine-tuning epochs using the whole 18,672 articles takes approximately 21 hours. In such cases in which the number of background documents available for the attacker outweigh its computational capabilities to train and conduct an exhaustive attack (and provided that the attacker does not know A_I), the strategy employed to define the *500_filtered* and *2000_filtered* scenarios would be the most reasonable choice. This reaffirms the fact that these two latter scenarios are more realistic than *50_eval* and *500_random*, which define worse-case (from a privacy perspective) but more ideal (from the perspective of the attacker) scenarios.

After defining B_D for each scenario, the corresponding B_D', C_D, $Ç_D$ and $Ç_D'$ sets needed to build the training and development sets should be created as detailed in Section 5.6. For B_D', the documents in B_D are anonymized using spaCy NER. After that, $Ç_D$ is defined as a subset of the abstracts corresponding to the bodies in B_D. Since the abstracts are summaries of the article bodies, this procedure can be considered equivalent to the summarization-based approach proposed in Section 5.6, which means that C_D does not need to be explicitly built. Finally, the same method employed for creating B_D' is applied to the documents in $Ç_D$. This results in the $Ç_D'$ set that constitutes the development set. The size of $Ç_D'$ was set to 30% for *50_eval*, and 10% for the *2000_filtered*, *500_filtered* and *500_random* scenarios.

6.3 Testing environment and hyperparameters

To simulate the implementation of TRIA by potential attackers, we selected Google Colaboratory[15] -a web-based IDE for interactive Python programming on the cloud- as execution environment. Our experiments were executed in a stable environment consisting of an Intel Xeon CPU, 12GB of RAM and an Nvidia Tesla K80 GPU with 16GB of VRAM.

As discussed in Section 5.1, a BERT-based classifier was employed to conduct the attack. We considered the popular pre-trained language models based on BERT outlined in Table 1. For the experiments varying the background knowledge, we opted for DistilBERT[16] [64], because it offers an outstanding trade-off between accuracy and computational cost.

The remainder of our implementation was performed as described in Sections 5.3, 5.4 and 5.5. Hyperparameters search was performed by using the re-identification accuracy at the development set as selection criteria, as outlined in Section 5.6. Ideally, this search should be performed independently for each B_D defined in the previous subsection. Nevertheless, given the number of tests to be conducted, their runtime, and the similarities of the scenarios, we only applied it to the *50_eval* scenario and then employed the obtained parameters for all the evaluated scenarios. Concretely, the hyperparameters that obtained the best accuracy at the development set were: *learning rate*=5e-5 (for both additional pre-training and fine-tuning), *batch size for additional pretraining*=8, *batch size for fine-tuning*=16, *sliding window length/overlap for additional pretraining*=512/128 and *sliding window length/overlap for fine-tuning*=100/25. Training relied on an Adam optimizer with *betas* set to 0.9 and 0.999. The additional pre-training was performed for 3 epochs and the fine-tuning for a maximum of 20 epochs. Using early stopping with a patience of 5 epochs and the accuracy on the development set as criteria, fine-tuning was run for ~20 epochs for the *50_eval*, *500_random* and *500_filtered* scenarios, and for ~10 epochs for the *2000_filtered* scenario.

6.4 Influence of attacker's resources

Hereafter we perform a comprehensive analysis of TRIA and TRIR according to the resources assumed to be available to attackers. According to the characterization of such resources introduced in Section 4.2, we considered:

- *Background knowledge breadth*: By using the different background sets presented in Section 6.2, we assessed the role of the size and type of the population in the background knowledge available to attackers.
- *Background knowledge depth*: We considered two different sizes for the background documents (i.e., article bodies): full bodies, which were used in most experiments, and the first 25% of the body contents, which were employed to analyze the influence of the background knowledge depth.
- *Computational capabilities*: We evaluated how the complexity of the attacker's inference affected TRIA's accuracy and runtime. This involved using different document splitting strategies (e.g., sliding windows of different sizes), usage of anonymized B_D (i.e., B'_D), and varying the size of the base language model employed to

[15] https://colab.research.google.com/

[16] We specifically relied upon the *distilbert-base-uncased* pre-trained model available on HuggingFace (https://huggingface.co/docs/transformers).

build the classifier (*i.e.*, DistilBERT, BERT or RoBERTa).

By configuring these aspects, we can set ideal (i.e., most powerful) or more feasible (but less powerful) attack scenarios.

6.4.1 Background knowledge breadth

Results depicted in Figure 4 show the influence of the different background knowledge breadths (corresponding to the B_Ds described in Section 6.2) in the re-identification of the anonymized documents (A_D) resulting from the methods introduced in Section 6.1. Notice that the "average" column in this and the forthcoming figures has been computed without considering the results for $A_D^{Clear\ text}$.

Results show that increasing background knowledge breadths significantly reduce the attack's accuracy, which is in line with the discussion in Section 4.2. The only exception is the *500_random* scenario that, despite having a B_I set much larger than A_I, resulted in just slightly less re-identification risk than *50_eval*. This was caused by the low similarity between the randomly selected individuals and the 50 protected ones, which made the latter easily distinguishable within the random set. In comparison, the risk of the filtered B_Ds (*500_filtered* and *2000_filtered*) was significantly lower because *i)* not all the protected subjects were present in B_D, and *ii)* those present were closer to the other individuals in B_D, thereby being more difficult to differentiate.

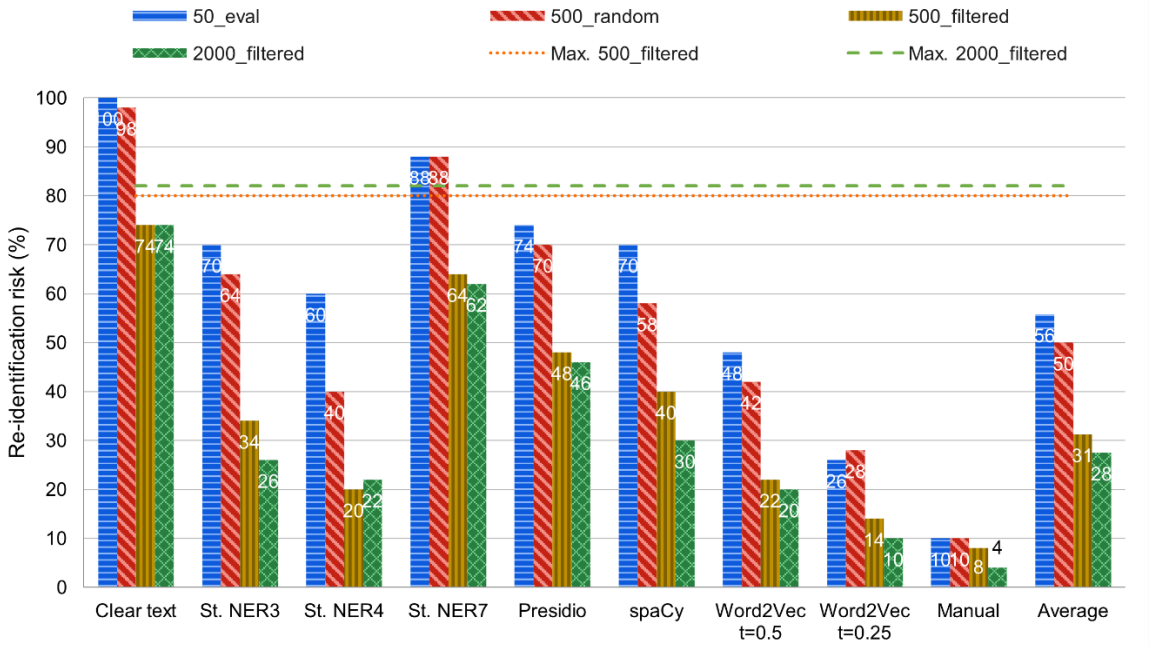


Figure 4: TRIR using different background knowledge breadths for several anonymization approaches. Horizontal lines depict the upper bounds for 500_filtered and 2000_filtered.

It is worth noting that the TRIR of $A_D^{Clear\ text}$ (*i.e.*, non-anonymized documents) is very close to the maximum risk of each scenario, being 100% for *50_eval*, 98% for *500_random* and 74% for *500_filtered* and *2000_filtered*. This demonstrates the capability of the additionally pre-trained and fine-tuned DistilBERT model as classifier for the re-identification task. For anonymized documents, TRIA was able to re-identify individuals even from A_D^{Manual}, with accuracies greater than the random guess, which were 2% for *50_eval*, 0.2% for *500_random*, 0.17% for *500_filtered* and 0.05% for *2000_filtered*. The fact that A_D^{Manual} did not achieve a perfect protection (*i.e.*, its re-identification accuracy was above random guess), highlights the susceptibility of human annotations to omissions and errors, as discussed in Section 1 and Section 3. This is caused by the difficulty of accounting for all the background knowledge and, therefore, all the (quasi-)identifying information (and their combinations) to be considered during manual anonymization.

Results for methods based on NER show weak protection, with re-identification risks greater than 45% for the *50_eval* worst-case scenario and, still, greater than or equal to 20% for *2000_filtered*. Results for A_D^{NER7} are the worst in all cases, probably due to the lack of a LOCATION and MISC types, which encompass a large variety of quasi-identifying information. In contrast, the Word2Vec instantiations achieved the lowest risks among all automated methods and across all B_Ds. As expected, $A_D^{Word2Vec\ t=0.25}$ produced a better protection than $A_D^{Word2Vec\ t=0.5}$ thanks to its stricter threshold. The risks figures of $A_D^{Word2Vec\ t=0.25}$ are also quite similar to those obtained by the manual anonymization. The outstanding protection of this method is the result of not limiting masking to a pre-defined set of categories (as NER-based methods do).

The runtime of TRIA mainly depends on the classifier fine-tuning step, which in turn is influenced by the number of samples. On this basis, 20 epochs of fine-tuning on *50_eval* took 30' while for *500_random* it required 1h 40'. For

500_filtered, the same 20 epochs took 4h 10', 2.5 times more than for *500_random*. This is in line with the length of the documents, which is 3 times larger for *500_filtered* than for *500_random*, because the popularity-related filters applied resulted in longer articles (*i.e.*, popular actors have longer articles). Lastly, 10 epochs with *2000_filtered* required 5h.

6.4.2 Background knowledge depth

To analyze the influence of the background knowledge depth available to attackers, we created a reduced version of the B_{DS} sets presented in Section 6.2 encompassing the first 25% of the content in the corresponding Wikipedia article bodies. Figure 5 depicts the results for all the anonymization methods with the depth-reduced background knowledge sets.

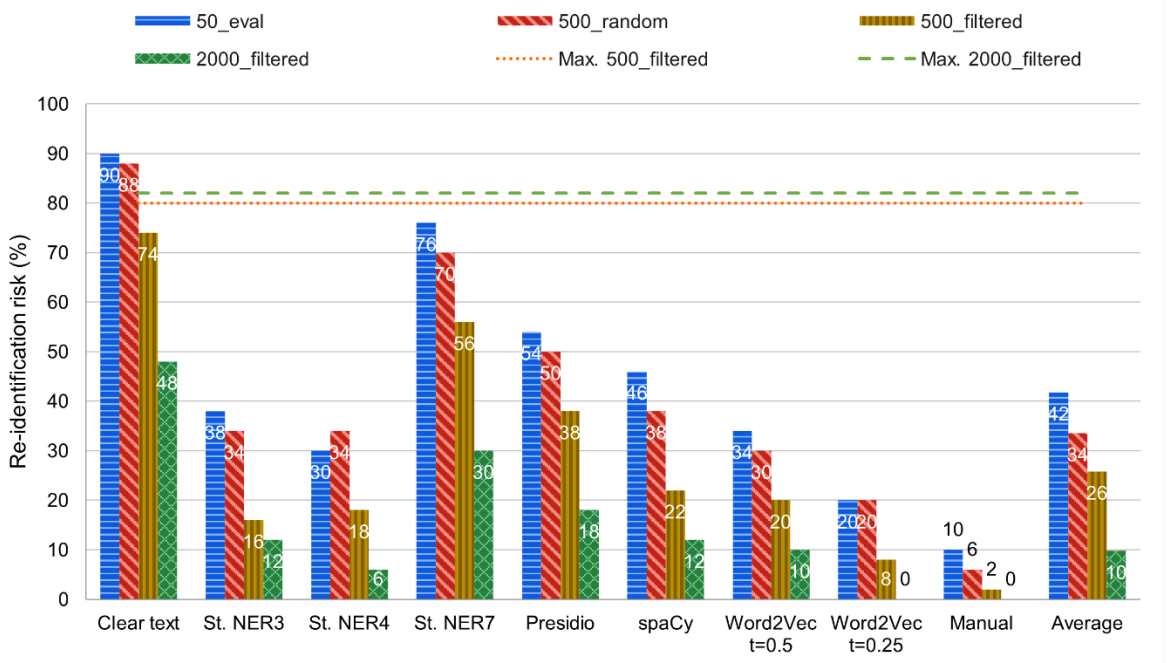


Figure 5: TRIR using the first 25% of the content from the background knowledge texts

Comparing these results with those in Figure 4 we observe significantly lower risk figures, with values near zero for the manual and Word2Vec approach in some configurations. Differences were more pronounced for the largest background knowledge set (*2000_filtered*), which indicates that, for larger background populations (breadth), the amount of background knowledge (depth) available is crucial to differentiate individuals and enable accurate re-identifications.

As the overall risk has decreased, the gap between the different automatic anonymization methods also narrowed. Nonetheless, their relative rankings remain largely unchanged. An exception to this trend is seen for A_D^{NER3} and A_D^{NER4}, which obtained risks comparable with those of $A_D^{Word2Vec\ t=0.5}$. This suggests that the quasi-identifiers missed by those NER-based methods (and exploited by TRIA) were mostly located beyond the first 25% of content of the background documents.

6.4.3 Computational capabilities

The most computational-intensive component of TRIA is building the classifier (see Section 5.4), particularly, during the fine-tuning step when the model is trained for re-identification. We next report experiments to evaluate how the different settings that can be tuned by attackers to improve training (and, therefore, improve inference quality at the cost of computation time) affect TRIA's accuracy and runtime. To maximize the observable differences to changes to those settings, all tests were conducted on the *50_eval* worst-case scenario, which is the least affected by the size of B_I and its divergence with A_I.

Figure 6 illustrates the effect of the splitting technique (detailed in Section 5.3) during the fine-tuning that, as stated in Section 5.5, is the same used for prediction. Sliding-window *N/M* defines a window of length *N* and an overlap *M* with the previous window. On the other hand, *sentence-based* stands for one split per sentence, an alternative splitting technique that produced splits with an average length of 23.5 tokens.

As shown on Figure 6, there is a general trend whereby smaller splits sizes produce more accurate results. This is in line with the hypothesis made in Section 5.4, which associated longer window lengths with overfitting. During the fine-tuning step, using long splits may result in samples too specific to the publicly available sources of the individual (*i.e.*, the article's body in B_D and its anonymized version in B_D'), while reducing the number of samples from which learn the class (since, in general, the longer the split, the fewer the number of splits available). Subsequently, the model is likely to overfit to the training documents rather than learning generalizable (quasi-)identifiers that facilitate re-identification. This makes it preferable to use shorter splits, but these would increase fine-tuning runtime. In fact, fine-tuning time scales almost

linearly to the number of splits and, therefore, runtimes are significantly larger for smaller split sizes (*i.e.*, 15,873 splits and 2h 30' for sentence-based, 4,217 splits and 30' for 100/25 sliding-window, and 2,088 and 15' for 200/50 sliding-window). For this reason, we used the sliding window 100/25 configuration in all the other tests as a well-balanced trade-off between fine-tuning time and accuracy.

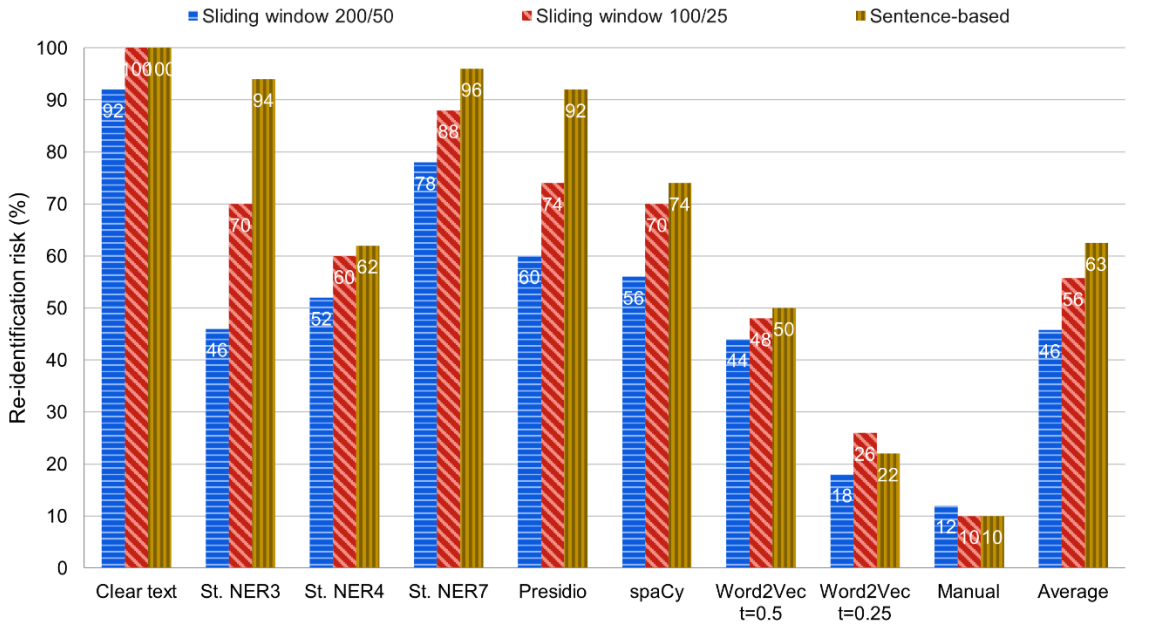


Figure 6: Comparison between different split sizes at fine-tuning for the *50_eval* scenario

Figure 7 presents an ablation study in which the classifier is only trained with the raw bodies B_D or the anonymized bodies B_D', instead of the combination of both sets (as we proposed in Section 5.4). These ablations almost halve the training data (and consequent runtime) of the additional pre-training and fine-tuning steps. Figure 7 shows that, in most cases, important benefits are obtained by using B_D' instead of B_D, and that their combination further improves the results. The improvements are considered worth the increase of training time (30' instead of 20'). This is in line with the arguments given in Section 5.4, probably with B_D' helping the model to deal with anonymized documents, and B_D providing the information corresponding to the masking replacements. Specifically, documents in B_D' allow the model to adapt to the data distribution of anonymized documents, which differs from that of B_D due to the masking replacements. Moreover, they allow the model to discover text spans neglected by the anonymization method that are also present in documents from A_D. When B_D and B_D' are combined, the model can learn useful information on how documents have been anonymized, thereby facilitating re-identification.

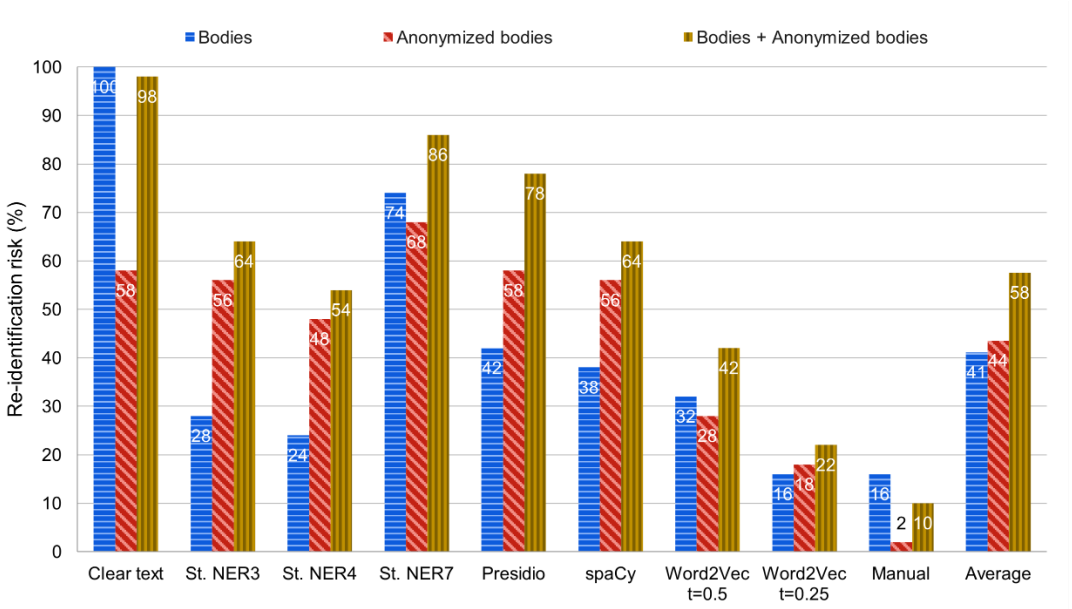


Figure 7: Ablation study on the inclusion of anonymized bodies in training for the *50_eval* scenario

Finally, Figure 8 illustrates the effect of the neural language model employed for the re-identification. We compare results obtained with DistilBERT (*distilbert-base-uncased*, with ~67M parameters) with those obtained by using the standard BERT model (*bert-base-uncased*, with ~110M parameters), and RoBERTa (*roberta-base*, with ~124M parameters). Notice that the combination of the narrowest knowledge breadth (*50_eval*), deepest background knowledge (complete article bodies) and largest model (RoBERTa) defines the most unconstrained and powerful attacker configuration considered so far. Fine-tuning runtimes in this experiment were 30' for DistilBERT, 1h for BERT and 1h 15' for RoBERTa. The accuracies averages show that, in comparison with DistilBERT, the improvement provided by BERT is small, whereas RoBERTa performed slightly worse for automated methods. The minor improvement of BERT is in line with the negligible loss of accuracy (~3%) observed by the authors of DistilBERT in other tasks [64]. The relatively poor performance of RoBERTa is, on the other hand, surprising. One possible explanation might come from the training data employed for its pre-training. All BERT, DistilBERT and RoBERTa use the Wikipedia as training data, which therefore matches our evaluation dataset; but, whereas for BERT and DistilBERT this represents 75% of the training data, for RoBERTa it is just 7.5%. Exceptionally, the re-identification accuracy for A_D^{Manual} seems to significantly benefit from the (usually) better-performing models (*i.e.*, RoBERTa>BERT>DistilBERT). This may be motivated by the stricter manual anonymization, which requires re-identification to focus on learning the few highly specific (quasi-)identifiers that may remain in the anonymized documents. In that case, re-identification depends more on the amount of knowledge available to build the model, rather than on Wikipedia-specific knowledge.

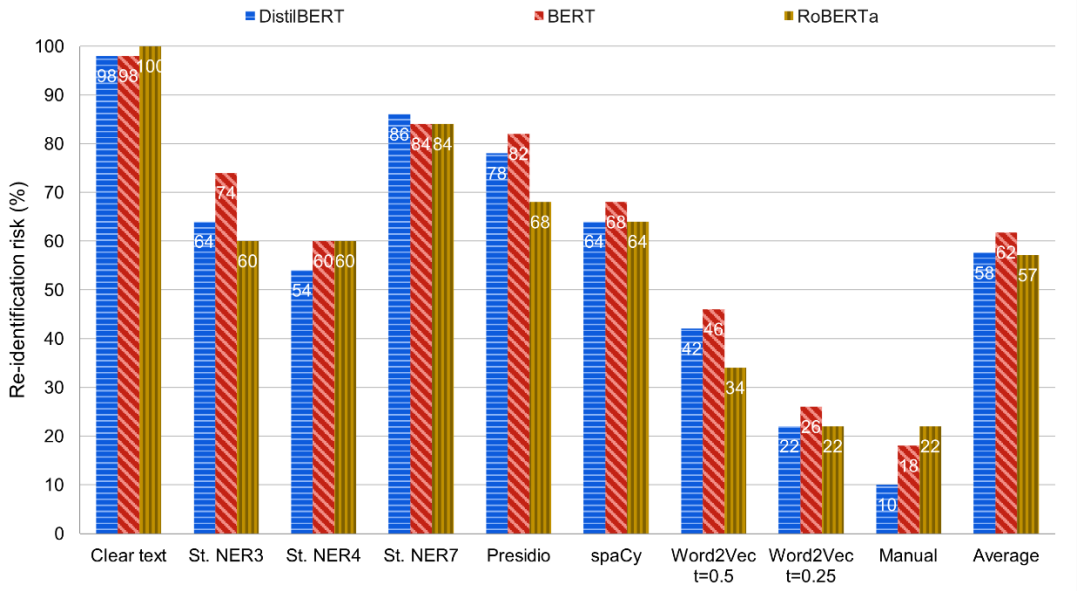


Figure 8: Effect of different language models on TRIR

6.5 Comparison with recall and human annotations

As mentioned in Section 3, the goal of empirical risk assessment is evaluating whether the protection level achieved by the anonymized documents fulfils the privacy requirements of the data holder. The recall metric (Eq. 2) employed in the literature assesses risk by comparing the automatic anonymization with manual annotations, which are considered the anonymization ground truth. Consequently, it evaluates a single (and presumably strict) privacy requirement encompassed by that manual ground truth. In contrast, TRIA (and, therefore, TRIR) can be configured for evaluating distinct privacy requirements according to the assumptions made on the attacker's resources. As shown in the previous section, by calibrating those resources, one may simulate ideal/powerful attackers that would define a worst-case scenario for privacy, or more feasible (and less powerful) attackers with more realistic access to data sources and computational capabilities.

In the following, we compare recall and TRIR under different attacker configurations with varying background knowledge breadths and depths that, as shown in the previous section, are the factors with the greatest influence in the risk assessment. To compute the recall for each method, we used its standard implementation[17] and the manual annotations from [32] as ground truth. Given that recall quantifies the inverse of the disclosure risk, we leveraged its complementary (*100% - Recall*) to enable a direct comparison with TRIR. The comparison was done in terms of the Pearson correlation between both metrics for the different anonymization methods considered. Results are reported in Table 2.

[17] https://github.com/NorskRegnesentral/text-anonymization-benchmark/blob/master/evaluation.py

Table 2: Correlation between recall and TRIR with different attacker's resources for the different anonymization methods.

Background knowledge breadth	100% background knowledge depth	25% background knowledge depth
50_eval	0.871	0.648
500_random	0.820	0.664
500_filtered	0.699	0.573
2000_filtered	0.690	0.694

Correlation figures vary substantially, with the highest value corresponding to the most powerful attacker configuration (*50_eval* background breadth and 100% background knowledge depth). This is consistent with the manual annotations conducted in [32], which were based on sound privacy-oriented annotation guidelines and that, as shown in the figures reported in the previous section, resulted in the strongest anonymization. Inversely, the lowest correlation corresponded to one of the weakest (but also more feasible) attacker configurations.

Figure 9 depicts a detailed comparison between recall and TRIR for the most and least correlated attacker configurations. We can see that, even though recall was strongly correlated with TRIR's most powerful attack scenario, it also systematically underestimated the observed re-identification risk. This is the result of recall relying on (imperfect) manual annotations, which exhibited a non-negligible re-identification risk (10%) under this attacker configuration. Differences with respect to the least correlated configuration were larger, which shows the inability of single annotations to encompass different privacy requirements or attack scenarios. These results showcase TRIR as an unsupervised alternative to recall, which is free from its dependencies (on human annotations) and that is more flexible to encompass varying privacy requirements.

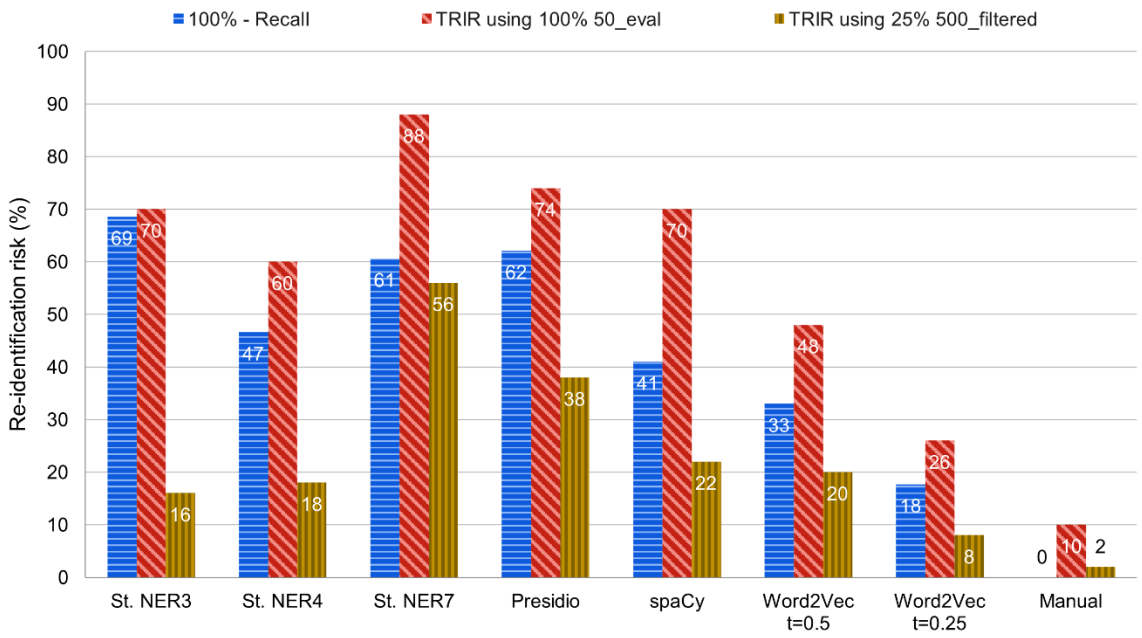


Figure 9: Comparison between the standard recall metric and the most and least correlated TRIR's attack configurations

As a complement to the former results, and to better understand the limitations of manual annotations, we also measured the *precision* of the automatic anonymization methods. Precision computes the percentage of automatically masked text spans that were also annotated by humans, and it is used in the literature as an estimate of the utility preserved by anonymized documents. A low precision indicates that terms were unnecessarily masked, thereby hampering the utility of the anonymized documents. Figure 10 reports precision values for the different methods using the same implementation as for recall.

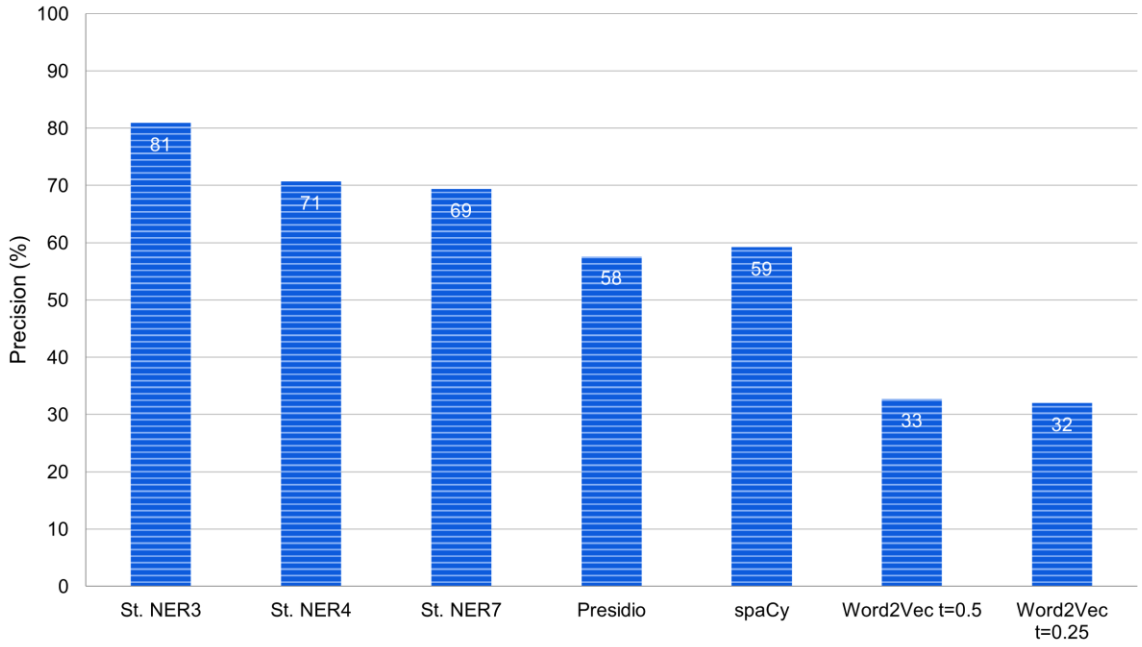


Figure 10: Precision of automatic anonymization methods

Consistently with the previous privacy assessments, there are notable differences between NER-based approaches and the Word2Vec-based method. On one hand, the higher precisions of NER-based methods derive from the fact that most of the (limited) NE types they are able to detect actually correspond to (quasi-)identifying information. Decrease of precision among these methods is also related to the diversity of named entity types each method encompasses, with broader coverage resulting in lower precision. Because this broader coverage not only did not provide better privacy, but resulted in the highest risk (see NER7, Presidio and spaCy in Figure 9), we can conclude that more extensive NER coverage does not necessarily lead to better anonymization.

On the other hand, the Word2Vec-based method resulted in the lowest precision. However, the fact that both Word2Vec instantiations obtained nearly identical precision but significantly different recall/TRIR, makes for an interesting analysis. First, this indicates that most of the identifying text spans detected in $A_D^{Word2Vec\ t=0.25}$ but not in $A_D^{Word2Vec\ t=0.5}$ were also detected by humans in A_D^{Manual}, thereby increasing recall. Those are the text spans whose embeddings obtained a similarity in the range (0.25, 0.5] with the individual's name, probably corresponding to slightly disclosive quasi-identifiers. Subsequently, the 67% of text spans detected in $A_D^{Word2Vec\ t=0.5}$ that were not detected by humans in A_D^{Manual}, obtained similarities in the range (0.5, 1] that, because their closeness to the individual's identity, are expected to correspond to highly disclosive quasi-identifiers. If the Word2Vec criterion is minimally correct (which it certainly seems to be because of its strong protection), this would imply that the human annotators missed some disclosive text spans. This observation aligns with the non-negligible risk measured for the manual anonymization in most attack configurations, and underscores the humans' difficulties to detect non-obvious (quasi-)identifiers that can be leveraged by attackers to conduct re-identifications.

7 Conclusions and future work

We have proposed two privacy evaluation tools for text anonymization: a *Text Re-Identification Attack* (TRIA) and an associated *Text Re-identification Risk* (TRIR) metric. Compared to the recall-based risk assessment ubiquitously employed in the text anonymization literature, our approach provides an unsupervised alternative that does not rely on costly manual annotations, and that is flexible enough to encompass a wide vary of privacy requirements/attack configurations. Moreover, whereas recall is a completeness measure that has been used as a proxy of true risk assessment, our metric measures privacy in the same manner it would be threatened: performing re-identification attacks by exhaustively exploiting the available background knowledge. Empirical results highlighted the limitations of recall due to its dependency on (imperfect) human annotations. This is consistent with limitations of the use of absolute recall values as a measure of protection/residual risk highlighted in recent studies [14, 39, 66]. Moreover, we provided additional empirical evidence to the criticisms raised in [14, 32] on the drawbacks of using NER methods for text anonymization.

As future work we plan to explore additional attack-oriented techniques, such as leveraging the confidence of the classifier predictions in order to improve re-identification precision. We also plan to leverage explainability techniques [78] on TRIA to identify the terms that greatest contributed to correctly classify/re-identify documents. These can then be used as feedback to improve the anonymization, thereby creating a virtuous anonymization cycle by which the risk assessment contributes to a gradual and systematic reduction of the privacy risk. We also plan to design a utility measure

in line with TRIR and alternative to the precision metric commonly employed in the literature. As with recall, IR-based precision suffers from several limitations when employed to measure the degree of utility preserved in the anonymized outcomes, as it as roughly equates the information loss resulting from anonymization to the number of false positives with respect to manual annotations [39]. Our idea is to propose a metric that, by also leveraging state-of-the-art language models, accurately measures the information loss resulting from *each* masking operation.

Acknowledgments. This research was funded by the European Commission (project H2020-871042 "SoBigData++"), the Norwegian Research Council (CLEANUP project, grant nr. 308904), MCIN/AEI/ 10.13039/501100011033 and "ERDF A way of making Europe" under grant PID2021-123637NB-I00 "CURLING", INCIBE and European Union NextGenerationEU/PRTR (project "HERMES" and INCIBE-URV Cybersecurity Chair) and the Government of Catalonia (ICREA Acadèmia Prize to D. Sánchez).

References

1. Meystre, S.M., Friedlin, F.J., South, B.R., Shen, S., Samore, M.H.: Automatic de-identification of textual documents in the electronic health record: a review of recent research. BMC Medical Research Methodology 10, 1-16 (2010)
2. Rivas, R., Hristidis, V.: Effective social post classifiers on top of search interfaces. Data Mining and Knowledge Discovery 35, 1809-1829 (2021)
3. Gutiérrez-Batista, K., Campaña, J.R., Vila, M.-A., Martin-Bautista, M.J.: Building a contextual dimension for OLAP using textual data from social networks. Expert Systems with Applications 93, 118-133 (2018)
4. Zhao, Y., Xu, X., Wang, M.: Predicting overall customer satisfaction: Big data evidence from hotel online textual reviews. International Journal of Hospitality Management 76, 111-121 (2019)
5. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data and Repealing Directive 95/46/EC. In: Commission, E. (ed.), (2016)
6. Domingo-Ferrer, J., Torra, V.: Privacy in data mining. Data mining and knowledge discovery 11, 117-119 (2005)
7. Bertino, E., Fovino, I.N., Provenza, L.P.: A framework for evaluating privacy preserving data mining algorithms. Data Mining and Knowledge Discovery 11, 121-154 (2005)
8. Agrawal, S., Haritsa, J.R., Prakash, B.A.: FRAPP: A framework for high-accuracy privacy-preserving mining. Data Mining and Knowledge Discovery 18, 101-139 (2009)
9. Hajian, S., Domingo-Ferrer, J., Monreale, A., Pedreschi, D., Giannotti, F.: Discrimination-and privacy-aware patterns. Data Mining and Knowledge Discovery 29, 1733-1782 (2015)
10. Li, N., Li, T., Venkatasubramanian, S.: t-closeness: Privacy Beyond k-Anonymity and l-Diversity. In: IEEE 23rd International Conference on Data Engineering, pp. 106-115. IEEE, (2007)
11. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M.: l-diversity: Privacy Beyond k-Anonymity. ACM Transactions on Knowledge Discovery from Data 1, 3-es (2007)
12. Samarati, P.: Protecting Respondents' Identities in Microdata Release. IEEE Transactions on Knowledge and Data Engineering 13, 1010-1027 (2001)
13. Dwork, C.: Differential Privacy. In: International Colloquium on Automata, Languages and Programming, pp. 1-12. Springer, (2006)
14. Lison, P., Pilán, I., Sánchez, D., Batet, M., Øvrelid, L.: Anonymisation Models for Text Data: State of the art, Challenges and Future Directions. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 4188-4203. Association for Computational Linguistics, (2021)
15. Csányi, G.M., Nagy, D., Vági, R., Vadász, J.P., Orosz, T.: Challenges and Open Problems of Legal Document Anonymization. Symmetry 13, 1490 (2021)
16. Bier, E., Chow, R., Gollé, P., King, T.H., Staddon, J.: The Rules of Redaction: Identify, Protect, Review (and Repeat). IEEE Security & Privacy 7, 46-53 (2009)
17. Aberdeen, J., Bayer, S., Yeniterzi, R., Wellner, B., Clark, C., Hanauer, D., Malin, B., Hirschman, L.: The MITRE Identification Scrubber Toolkit: Design, training, and assessment. International Journal of Medical Informatics 79, 849-859 (2010)
18. Chen, A., Jonnagaddala, J., Nekkantti, C., Liaw, S.-T.: Generation of Surrogates for De-Identification of Electronic Health Records. MEDINFO 2019: Health and Wellbeing e-Networks for All, pp. 70-73. IOS Press (2019)
19. Dernoncourt, F., Lee, J.Y., Uzuner, O., Szolovits, P.: De-identification of patient notes with recurrent neural networks. Journal of the American Medical Informatics Association 24, 596-606 (2017)
20. Elazar, Y., Goldberg, Y.: Adversarial Removal of Demographic Attributes from Text Data. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 11-21. Association for Computational Linguistics, (2018)
21. Hassan, F., Domingo-Ferrer, J., Soria-Comas, J.: Anonymization of Unstructured Data Via Named-Entity

Recognition. In: International Conference on Modeling Decisions for Artificial Intelligence, pp. 296-305. Springer, (2018)
22. Huang, Y., Song, Z., Chen, D., Li, K., Arora, S.: TextHide: Tackling Data Privacy in Language Understanding Tasks. In: Findings of the Association for Computational Linguistics: EMNLP 2020. Association for Computational Linguistics, (2020)
23. Johnson, A.E., Bulgarelli, L., Pollard, T.J.: Deidentification of free-text medical records using pre-trained bidirectional transformers. In: Proceedings of the ACM Conference on Health, Inference, and Learning, pp. 214-221. Association for Computing Machinery, (2020)
24. Liu, Z., Tang, B., Wang, X., Chen, Q.: De-identification of clinical notes via recurrent neural network and conditional random field. Journal of Biomedical Informatics 75, S34-S42 (2017)
25. Mamede, N., Baptista, J., Dias, F.: Automated anonymization of text documents. In: IEEE Congress on Evolutionary Computation, pp. 1287-1294. IEEE, (2016)
26. Neamatullah, I., Douglass, M.M., Lehman, L.-W.H., Reisner, A., Villarroel, M., Long, W.J., Szolovits, P., Moody, G.B., Mark, R.G., Clifford, G.D.: Automated de-identification of free-text medical records. BMC Medical Informatics and Decision Making 8, 1-17 (2008)
27. Reddy, S., Knight, K.: Obfuscating Gender in Social Media Writing. In: Proceedings of the First Workshop on NLP and Computational Social Science, pp. 17-26. Association for Computational Linguistics, (2016)
28. Szarvas, G., Farkas, R., Busa-Fekete, R.: State-of-the-art Anonymization of Medical Records Using an Iterative Machine Learning Framework. Journal of the American Medical Informatics Association 14, 574-580 (2007)
29. Xu, Q., Qu, L., Xu, C., Cui, R.: Privacy-Aware Text Rewriting. In: Proceedings of the 12th International Conference on Natural Language Generation, pp. 247-257. Association for Computational Linguistics, (2019)
30. Yang, H., Garibaldi, J.M.: Automatic detection of protected health information from clinic narratives. Journal of Biomedical Informatics 58, S30-S38 (2015)
31. Yogarajan, V., Mayo, M., Pfahringer, B.: A survey of automatic de-identification of longitudinal clinical narratives. arXiv preprint arXiv:1810.06765 (2018)
32. Hassan, F., Sanchez, D., Domingo-Ferrer, J.: Utility-Preserving Privacy Protection of Textual Documents via Word Embeddings. IEEE Transactions on Knowledge and Data Engineering (2021)
33. Sánchez, D., Batet, M.: C-sanitized: A privacy model for document redaction and sanitization. Journal of the Association for Information Science and Technology 67, 148-163 (2016)
34. Mosallanezhad, A., Beigi, G., Liu, H.: 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, pp. 2360-2369. Association for Computational Linguistics, (2019)
35. Chakaravarthy, V.T., Gupta, H., Roy, P., Mohania, M.K.: Efficient Techniques for Document Sanitization. In: Proceedings of the 17th ACM conference on Information and knowledge management, pp. 843-852. Association for Computing Machinery, (2008)
36. Fernandes, N., Dras, M., McIver, A.: Generalised Differential Privacy for Text Document Processing. In: International Conference on Principles of Security and Trust, pp. 123-148. Springer, (2019)
37. Cumby, C., Ghani, R.: A Machine Learning Based System for Semi-Automatically Redacting Documents. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 1628-1635. AAAI Press, (2011)
38. Anandan, B., Clifton, C., Jiang, W., Murugesan, M., Pastrana-Camacho, P., Si, L.: t-Plausibility: Generalizing Words to Desensitize Text. Transactions on Data Privacy 5, 505-534 (2012)
39. Pilán, I., Lison, P., Øvrelid, L., Papadopoulou, A., Sánchez, D., Batet, M.: The Text Anonymization Benchmark (TAB): A Dedicated Corpus and Evaluation Framework for Text Anonymization. Computational Linguistics 1-49 (2022)
40. Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Nordholt, E.S., Spicer, K., De Wolf, P.-P.: Statistical Disclosure Control. Wiley New York, New York (2012)
41. Domingo-Ferrer, J., Torra, V.: Disclosure risk assessment in statistical microdata protection via advanced record linkage. Statistics and Computing 13, 343-354 (2003)
42. Nin Guerrero, J., Herranz Sotoca, J., Torra i Reventós, V.: On method-specific record linkage for risk assessment. In: Proceedings of the Joint UNECE/Eurostat Work Session on Statistical Data Confidentiality, pp. 1-12. UNECE, (2007)
43. Torra, V., Abowd, J.M., Domingo-Ferrer, J.: Using Mahalanobis Distance-Based Record Linkage for Disclosure Risk Assessment. In: Privacy in Statistical Databases, pp. 233-242. Springer, (2006)
44. Torra, V., Stokes, K.: A formalization of record linkage and its application to data protection. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 20, 907-919 (2012)
45. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: 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. Association for Computational Linguistics, (2018)
46. Manzanares-Salor, B., Sánchez, D., Lison, P.: Automatic Evaluation of Disclosure Risks of Text Anonymization Methods. In: Privacy in Statistical Databases, pp. 157-171. Springer, (2022)
47. Mackey, E., Elliot, M., O'Hara, K.: The Anonymisation Decision-making Framework. UKAN Publications, Manchester (2016)

48. Chevrier, R., Foufi, V., Gaudet-Blavignac, C., Robert, A., Lovis, C.: Use and Understanding of Anonymization and De-Identification in the Biomedical Literature: Scoping Review. Journal of Medical Internet Research 21, e13484 (2019)
49. Abril, D., Navarro-Arribas, G., Torra, V.: Improving record linkage with supervised learning for disclosure risk assessment. Information Fusion 13, 274-284 (2012)
50. Abril, D., Torra, V., Navarro-Arribas, G.: Supervised learning using a symmetric bilinear form for record linkage. Information Fusion 26, 144-153 (2015)
51. Batet, M., Sánchez, D.: Semantic Disclosure Control: semantics meets data privacy. Online Information Review 42, 290-303 (2018)
52. Sánchez, D., Batet, M.: Toward sensitive document release with privacy guarantees. Engineering Applications of Artificial Intelligence 59, 23-34 (2017)
53. Staddon, J., Golle, P., Zimny, B.: Web-Based Inference Detection. In: USENIX Security Symposium, pp. 1-16. Association for Computing Machinery, (2007)
54. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. In: International Conference on Learning Representations. Association for Computational Linguistics, (2013)
55. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.J.J.M.L.R.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research 21, 1-67 (2020)
56. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A.: Language Models are Few-Shot Learners. In: Advances in Neural Information Processing Systems, pp. 1877-1901. Neural Information Processing Systems Foundation, (2020)
57. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E.J.a.p.a.: On the opportunities and risks of foundation models. Radiology: Artificial Intelligence 4, e220119 (2021)
58. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention Is All You Need. In: Advances in Neural Information Processing Systems, pp. 5998-6008. Neural Information Processing Systems Foundation, (2017)
59. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation 9, 1735-1780 (1997)
60. Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the Properties of Neural Machine Translation: Encoder-decoder Approaches. In: Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, pp. 103–111. Association for Computational Linguistics, (2014)
61. Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., Fidler, S.: Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 19-27. IEEE, (2015)
62. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M.: Transformers: State-of-the-Art Natural Language Processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38-45. Association for Computational Linguistics, (2020)
63. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692 (2019)
64. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
65. Sun, X., Li, X., Li, J., Wu, F., Guo, S., Zhang, T., Wang, G.: Text Classification via Large Language Models. pp. 8990-9005. Association for Computational Linguistics, (2023)
66. Mozes, M., Kleinberg, B.: No intruder, no Validity: Evaluation Criteria for Privacy-Preserving Text Anonymization. arXiv preprint arXiv:2103.09263 (2021)
67. Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., Liu, Q.: TinyBERT: Distilling BERT for Natural Language Understanding. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 4163–4174. Association for Computational Linguistics, (2019)
68. Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., Zhou, M.: MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In: Advances in Neural Information Processing Systems, pp. 5776-5788. Association for Computing Machinery, (2020)
69. Rasmy, L., Xiang, Y., Xie, Z., Tao, C., Zhi, D.: Med-BERT: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. NPJ digital medicine 4, 86 (2021)
70. Mackenzie, J., Benham, R., Petri, M., Trippas, J.R., Culpepper, J.S., Moffat, A.: CC-News-En: A large English news corpus. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 3077-3084. (2020)
71. Gokaslan, A., Cohen, V.: URL http://web. archive. org/save/http://Skylion007. github. io. OpenWebTextCorpus
72. Trinh, T.H., Le, Q.V.: A simple method for commonsense reasoning. arXiv preprint arXiv:1806.02847 (2018)
73. Pappagari, R., Zelasko, P., Villalba, J., Carmiel, Y., Dehak, N.: Hierarchical transformers for long document

classification. In: IEEE Automatic Speech Recognition and Understanding Workshop, pp. 838-844. IEEE, (2019)
74. Sun, C., Qiu, X., Xu, Y., Huang, X.: How to Fine-Tune BERT for Text Classification? In: China National Conference on Chinese Computational Linguistics, pp. 194-206. Springer, (2019)
75. El-Kassas, W.S., Salama, C.R., Rafea, A.A., Mohamed, H.K.: Automatic text summarization: A comprehensive survey. Expert Systems with Applications 165, 113679 (2021)
76. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D.: The Stanford CoreNLP Natural Language Processing Toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55-60. Association for Computational Linguistics, (2014)
77. Weischedel, R., Hovy, E., Marcus, M., Palmer, M., Belvin, R., Pradhan, S., Ramshaw, L., Xue, N.: OntoNotes: A large training corpus for enhanced processing. In: Olive, J., Christianson, C., McCary, J. (eds.) Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation, pp. 54-63. Springer, New York (2011)
78. Lundberg, S.M., Lee, S.-I.: A Unified Approach to Interpreting Model Predictions. In: Advances in Neural Information Processing Systems, pp. 4768–4777. Association for Computing Machinery, (2017)