Utility-Preserving Privacy Protection of Textual Documents via Word Embeddings

Fadi Hassan, David Sánchez and Josep Domingo-Ferrer, *Fellow, IEEE*

Abstract—Text is the most usual way to share information in society. Yet, if textual documents contain personal sensitive information, they cannot be shared with third parties or released publicly without adequate protection. Privacy-preserving mechanisms provide ways to sanitize data so that identities and/or confidential attributes are not disclosed. In the last twenty years, a great variety of mechanisms have been proposed to protect structured databases with numerical and categorical attributes; however, little attention has been devoted to unstructured textual data. In general, textual data protection requires first detecting pieces of text that may lead to disclosure of sensitive information and then masking those pieces via suppression or generalization. Current solutions rely on pre-trained classifiers that can recognize a fixed set of (allegedly disclosive) named entities, such as names or locations. Yet, such approaches fall short of providing adequate protection because in reality disclosive information is not limited to a predefined set of entity types, and not all the appearances of certain entity type result in disclosure. Besides, named entity recognition requires considerable manual effort to tag the training data needed to build classifiers. In this work we propose a more general and flexible solution for textual data protection. By means of word embeddings we build vectors that numerically capture the semantic relationships of the textual terms appearing in a collection of documents. Then we evaluate the disclosure caused by the textual terms on the entity to be protected (*e.g.*, an individual's identity or a confidential attribute) according to the similarity between their vector representations. Our method also limits the semantic loss (and, therefore, the utility loss) of the document by replacing (rather than just suppressing) disclosive terms with privacy-preserving generalizations. Empirical results show that our approach offers much more robust protection and greater utility preservation than methods based on named entity recognition, with the additional important advantage of avoiding the burden of manual data tagging.

Index Terms—Privacy protection, textual documents, word embeddings, named entity recognition, redaction

1 Introduction

Text constitutes the most usual way to share information among human beings. Textual data are therefore a crucial resource for many businesses and researchers. For instance, medical histories and clinical notes are needed in medical and pharmacological research [29], publications in social networks can drive socioeconomic studies [2], or written opinions and reviews can be used to improve recommender systems [24]. However, textual data may carry sensitive personal information; in this is the case, they cannot be shared with third parties or released in the public sphere without properly protecting the fundamental right to privacy [55] of the individuals to whom the text refers.

Even though a panoply of privacy protection methods have been proposed in the literature [23], most of them focus on structured data (that is, data that conform to a regular model such as a database schema) and more concretely on numerical attributes [7]. This contrasts with the fact that the vast majority of data generated nowadays are unstructured [48], [54]. Specifically, unstructured text is the most common form of unstructured data, and it can be found in books, articles, web pages, emails, posts in social networks or clinical reports.

To protect structured databases, attributes are categorized according to their potential disclosure on the individual to whom a record corresponds. An *identifier* is an attribute whose values are enough to re-identify the individual to whom a record corresponds, whereas *quasi-identifiers* are attributes that separately do not allow re-identification but whose combination may. Both types of attributes entail *identity disclosure risk*. On the other hand, confidential attributes are those that may disclose sensitive information on the individual, thereby entailing *attribute disclosure risk*. The usual approach to data protection is to remove identifiers and mask quasi-identifiers (where masking can be enforced via perturbation, generalization or even suppression of values) [40]. While identifier attributes are usually easy to recognize, quasi-identifiers and confidential attributes are not. In general, we should classify as quasi-identifiers any set of attributes whose combined values may be available in an external data source that associates them with an identity.

If dealing with structured data may be challenging, protecting unstructured text is even more complex. First, we no longer have a fixed list of attributes: textual data may contain any information, which varies across documents. Furthermore, deciding what is a quasi-identifier or a confidential value is much more complex than with structured data: for each piece of text we need to judge whether it can be used for re-identification or may disclose sensitive values. Such a judgment is not easy for a human expert [8], let alone for a computer program.

In general, accurate protection of textual documents remains a largely manual process [8]. At most, (semi)automatic tools based on named entity recognition (NER) have been designed to remove –some– of the burden from the human experts. These tools are configured to pinpoint predefined entity types that are assumed to facilitate the re-identification of individuals (such as names, locations or dates).

The authors are with the UNESCO Chair in Data Privacy, CYBERCAT-Center for Cybersecurity Research of Catalonia, Department of Computer Engineering and Mathematics, Universitat Rovira i Virgili, Av. Països Catalans 26, E-43007 Tarragona, Catalonia, e-mail {fadi.hassan,david.sanchez,josep.domingo}@urv.cat.

However, NER-based techniques have important limitations. First, for the more sophisticated NER techniques one needs to train the classifiers, and this requires a large amount of manually tagged training data that match the language of the text to be protected. Tagging a sufficient volume of training data may become a considerable effort. Second, NER-based methods are unable to discern whether the pinpointed entities refer to the individual to be protected or not. Hence, systematic masking (for example suppression) of those entities often degrades the text semantics (and therefore its utility) without a corresponding reduction of risk. Finally, while NER systems detect a fixed set of entity types, there are unlimited ways of referring to (quasi-)identifying information in a text. As a result, NER-based methods are usually characterized by a low detection recall, which yields poorly protected outcomes.

In this work, we overcome the above-mentioned limitations of NER-based techniques by proposing a more general and flexible method that better captures the notion of disclosure risk as understood in the literature on data privacy [56]. We characterize the semantic relationships between the textual entities appearing in a document by leveraging word embeddings [32]. Word embeddings learn detailed vector representations of linguistic terms that convey the semantics of such terms. We make use of these vectors to measure the semantic relatedness and, from it, the extent to which the terms appearing in the text document disclose the entity to be protected. The latter can be either an individual's identity (*e.g.*, a name) or a confidential attribute (*e.g.*, a sensitive disease). Thus, our method naturally encompasses the notions of identity and attribute disclosure, and it automatically classifies the textual terms as being disclosive or not disclosive; that is, it automatically determines which terms act as (quasi-)identifiers of the entity to be protected. This delivers a more powerful solution to protect textual documents than NER-based methods, because our solution is not restricted to detecting predefined (quasi-)identifying types (*e.g.*, names or locations) and it can limit the protection only to the terms referring to the entity to be protected, whatever it is. The empirical results we report show that our method offers more robust protection than NER-based approaches. Regarding ease of use and deployment, our solution is mostly *language-agnostic* and does *not require manual tagging* of training data.

Beyond accurately meeting the privacy requirements, we also improve the masking of quasi-identifying terms in order to increase utility preservation. In contrast to approaches that simply suppress quasi-identifying terms, we propose a generalization-based masking procedure that preserves their underlying semantics as much as allowed by the privacy requirements, which state the maximum level of allowed disclosure. To this end, we rely on the taxonomies contained in structured knowledge bases that model the domains to which the entities appearing in the document belong. As we also show in the empirical work, with this approach we significantly reduce the information loss incurred by data masking in comparison with approaches based on data removal or NER-based classification of entities.

The rest of this paper is organized as follows. In Section 2 we review related work on textual document protection. In Section 3 we present our approach to document protection based on word embeddings. Section 4 contains an empirical evaluation of our method and a comparison against related contributions. In Section 5 we discuss several practical applications of our method that sustain its generality. Section 6 gathers conclusions and future work directions.

This paper is an extension of the preliminary research reported in the conference paper [22], which strictly focused on preventing re-identification. In contrast, here we extend this previous work by considering a broader notion of data protection, against both *identity* and *attribute* disclosure; furthermore, we propose a utility-preserving procedure to mask quasi-identifying terms. The abstract and the introduction have been significantly revamped to reflect this greater ambition. Sections 2 and 3 have been substantially expanded (including new figures) to consider newer embedding models and related works. Section 3.2.3 is entirely new. Further, the experiments in Section 4 have also been greatly augmented by (i) including utility metrics, additional NER and word embedding models and additional training corpora, (ii) evaluating the influence of the data pre-processing and disclosure thresholds and (iii) evaluating the actual protection offered by the different approaches against a machine learning-based re-identification attack. Finally, Section 5 is new and Section 6 has been rewritten.

2 Related Work

The task of protecting the private information of the individuals mentioned in text documents is referred to in the literature as *document redaction* [8], *sanitization* [44] or *anonymization* [5]. Whatever the name, it consists of two steps: (i) detecting (potentially) disclosive pieces of text, and (ii) masking those pieces appropriately.

For many years textual data protection has been a highly manual process [35], and it still is. Usually, several human experts review the text and mask all items they deem usable to re-identify individuals and/or disclose confidential data on them [8]. To reduce the burden of human experts, some systems that make use of NER have been introduced.

NER was created as a way to extract structured information, like person and organization names, locations, times or dates, from an unstructured text. Early NER systems were based on handcrafted rules or regular expressions. For instance, times can be identified using the following pattern: "at" + digits + "am"/"pm". Up until 2000, handcrafted rule systems offered the best results. Statistical approaches subsequently took over. In statistical NER systems, models such as HMM (hidden Markov models) or CRF (conditional random fields) are trained to locate a specific type of entity. With the development of deep learning neural networks, recurrent neural networks (RNN) and extensions of them such as long short-term memory (LSTM) and gated recurrent units (GRUs) surpassed the accuracy of statistical NER systems. Nowadays, the state of the art is based on transformers like BERT [15] and ELMo [37]. These are pre-trained on large amounts of data and, unlike previous models, they characterize words according to their context. Even though they are general-purpose NLP models, these contextual models can be tailored or fine-tuned to solve multiple tasks including NER, but also sentiment analysis, text generation, question answering, summarization or machine translation.

Training an NER model from scratch or tailoring an NLP model for NER require a considerable amount of tagged data that match the language to which the NER model is to be applied. Well-trained NER models usually have high precision (typically above 80%). Additionally, there are quite a few software packages available to carry out NER tasks, such as spaCy [50] or the Stanford NER [28].

Current solutions for textual data protection employ NER because they assume that the named entities (NEs) are the ones that entail the highest disclosure risk, as they refer to real-world entities. Amazon's Macie [4] locates several personally identifiable information items (like names, addresses, birthdates, etc.) and classifies documents in several categories according to their risk. Additionally, Macie is capable of detecting many information items that are regarded as confidential (like passwords, bank accounts, etc). Google's Cloud DLP [13] also leverages rules and machine learning techniques to detect the presence of confidential and re-identifying pieces of information. Similarly, Symantec's Data Loss Prevention [53] uses dictionaries and rules (to detect several types of information items that have a regular structure) as well as machine learning (to detect other types of identifiable and confidential information that lack a regular structure). Microsoft's Presidio tool [30] is based on combining regular expression matching, spaCy NER models and Flair [3] with BERT embeddings. It is trained on 80,000 samples generated with data augmentation techniques and can detect 17 (quasi-)identifying and confidential categories.

As evidenced by the number of commercial tools available, NER-based systems are practical enough to be employed in real-world applications. However, since they assume that all (and only) the NEs in a given document should be protected, they suffer from the severe limitations highlighted in the introduction. First, nouns or phrases other than NEs may also be (quasi-)identifying, such as demographic attributes or healthcare conditions. Second, not all the NEs appearing in a document should be protected, perhaps because they are very general entities (such as countries or large cities) or because they do not refer to the individual to be protected (as it may happen when the text refers to other individuals in addition to the one to be protected). Third, only the NE classes for which the classifier has been trained can be detected; this means that usually a few dozens of NE classes can be detected, whereas there is an unbounded number of potential quasi-identifying information types that may not resemble NEs at all (*e.g.*, demographic information may be quasi-identifying in some contexts). In summary, NER-based protection introduces considerable burden (due to the need of training), and typically results in both unnecessary masking and weak protection.

The methods and tools reviewed above solely focus on detecting (quasi-)identifying information or at most they suppress disclosive items or replace them with coarse NE values (like "person", "location", "date"). This falls short of optimizing data protection, which consists in using the minimum amount of masking required to meet the privacy requirements. The analytical utility of the protected outcomes ought to be preserved as much as possible, for them to be usefully shareable. Generalization is the most common utility-preserving masking technique applied to the protection of text [45]. Unlike other methods in the literature, such as entity swapping [1] and noise addition [20], generalization outputs truthful data [6], [11], [14]. The methods in [11] and [14] use generalization similarly to the way k-anonymity [40] is employed in structured databases: they assume a large and homogeneous collection of documents and generalize the quasi-identifying terms so that there are at least k identical generalizations in the collection. In this way, each disclosive term becomes indistinguishable from at least $k-1$ other terms in the collection. However, assuming a homogeneous set of documents and protecting them groupwise is quite restrictive. An approach that can individually sanitize documents is presented in [6]. The authors employ a knowledge base to generalize quasi-identifying terms so that at least t plausible versions of the generalized document can be created by combining specializations of the generalized terms. The authors acknowledge that setting the value of t is not intuitive and that it is hard to predict the protection offered by a concrete value because it depends on several factors including the document size, the number of terms to be masked and the detail of the knowledge base used to generalize terms.

The methods cited in the previous paragraph concentrate on masking quasi-identifying terms, but they assume those items have been already detected. An integral approach considering both detection and utility-preserving masking is presented in [42], [43], [44], [45]. The authors propose a privacy model grounded on information theory that quantifies disclosure risks as a function of the mutual information shared among the entities referred to in the document. Afterwards, quasi-identifying items are generalized so that the amount of information they disclose on the entity to be protected is sufficiently decreased. Even though this approach is more general than NER-based methods, it suffers from the need to compute accurate conditional probabilities among all the combinations of terms in the document. This hampers scalability to deal with large collections of documents.

Some authors have recently proposed privacy-preserving methods for text documents that build on word embeddings [19], [20], [26]. However, these works focus on obfuscating the authorship of the document, rather than protecting the privacy of the individuals referred to in the text. The authorship of a document and the author's attributes are inferred from the linguistic and stylistic properties/regularities of written text rather than the document's topic or the text semantics. Hence, the approaches to protecting the document author rely on distorting the distribution of words in the text via differentially private noise added to the word embeddings [19], [20] or on constraining the training of the embeddings to prevent disclosing certain attributes [26]. Thus, the outputs of those systems are –distorted– word distributions (*e.g.*, bag-of-words) [19] or or constrained embedding models [26] rather than actual documents. As a result, the outputs lose their readability and are only useful for applications employing these deconstructed representation of documents, such as topic classification. Finally, as discussed above, noise-based approaches in which words are probabilisticaly replaced by other words do not preserve the truthfulness of the output, unlike generalization-based masking, which is the usual approach to document sanitization.

3 Document Protection via Word Embeddings

The most widely accepted definition of privacy rests on the notion of informational self-determination, that is, "the claim of individuals, groups or organizations to determine for themselves when, how, and to what extent information about them is communicated to others" [56]. Following this definition, the crux of protecting data releases is the ability to detect (and subsequently remove or mask) the information that refers to a single entity and to no other entity. In other words, protecting one entity should not encroach on how another entity is protected. This is exactly the goal that our approach sets out to achieve. As discussed in the previous section, approaches based on NER fail in this respect because they implicitly assume the entire content of each document refers to a single entity.

To reach our goal, we need a way to characterize the textual terms according to the information they disclose on the entity to be protected. A common metric to quantify this "amount of disclosure" is the semantic relatedness between the terms in the document and the entity to be protected [44]. Traditionally, the semantic relatedness between linguistic entities has been assessed using distributional [33] or probabilistic models [41], which require accurate statistics on the (co-)occurrence of words. A recent trend in computational linguistics to measure the relatedness between words is to use word embedding models.

3.1 Background on Word Embeddings

Word embeddings map words into high-dimensional numerical vectors capturing their semantics. Word embedding models can be categorized into two main types: static embedding and dynamic/contextual embedding. Models like word2vec [31], fastText [9] and GloVe [36] are static and context-independent models: they build vector representations of words that do not depend on the context in which words appear. Word2vec uses a neural network [31] trained either to predict the current word from a window of neighboring words (continuous bag-of-words model) or to predict neighboring words based on the current word (skip-gram model). FastText [9] also works on the same idea, but the main difference is that fastText takes care of the out of vocabulary (OOV) problem by taking into consideration the subword information. Finally, GloVe [36] uses two methods to generate word representations: local context window information and aggregated global word-word co-occurrence statistics from the pre-training corpus. Unlike word2vec and fastText, GloVe does not rely just on local context window information, but incorporates global statistics to obtain a more accurate word representation.

However, the current state of the art is contextual/dynamic embedding with models like BERT [15] or ELMo [37]. These models are built using transformer-based self-supervised architectures that are pre-trained for language understanding. The key idea of these models is that the pre-training task is designed to be a generic form that can be tailored to solve any specific problem in NLP. Pre-training can be performed by using masked language models (MLM) or next-sentence prediction (NSP). MLM randomly masks some of the tokens from the input and the objective is to predict the original words, whereas NSP consists in identifying consecutive sentences. Both tasks aim at steering the model into taking the context of a word into consideration.

3.2 Our Approach

From the perspective of *distributional semantics*, words that are likely to co-occur in a context (or, otherwise put, those with similar contexts) tend to be semantically related [39]. Therefore, after training a word embedding model with a collection of word contexts, semantically related words will have similar word embedding vectors. The distributional semantics captured by word embedding models also encompasses a very broad notion of relatedness. Moreover, the larger the amount of data used, the more general the resulting distributional model [10]; in fact, word embedding models owe their success to the massive amount of data they use for training. On the other hand, a strong semantic relatedness between the words appearing in a text and the entity to be protected is what enables the semantic inferences that may lead to disclosure [5], [44]. Therefore, we propose to measure the disclosure risk caused by each term appearing in a document w.r.t. an entity to be protected as a function of the similarity between their word embedding vectors.

Our approach consists of three phases. In the first phase, we use a large corpus to train a word embedding model tailored to capture the semantic relationships that may cause disclosure. The trained model has learned the relationships (and, therefore, the pairwise disclosure risks) between all the terms appearing in the collection of documents. In the second phase, for a given document D, an entity to be protected e and a threshold t stating the maximum level of allowed disclosure, we use the trained model to detect the terms in D that may act as (quasi-)identifiers of e. Both e and t define the privacy requirements. In the third phase we mask the quasi-identifying terms we detected in the second phase. Masking is performed by replacing those terms by generalizations extracted from structured knowledge bases modeling the concepts of the domain. The generalizations are picked so that they are the most specific ones that are 'safe' (*i.e.*, non-disclosive enough) according to the risk criterion employed in the second phase. In this way, we protect privacy while retaining the semantics (and, therefore, the utility and readability) of document D as much as possible.

3.2.1 *Training the Model*

The first phase of our method is depicted in Figure 1. It has the following steps, which are explained further below:

- Data collection and pre-processing;
- Model training.

To train a word embedding model that accurately characterizes the disclosure-enabling semantic relationships affecting an entity or a set of entities, we need a representative "core" corpus of documents that describe those entities.

Ideally, the corpus ought to contain all the documents that shall be protected (for instance, a collection of medical records). In this way we ensure that all the terms in such documents appear in the model's vocabulary and get accurate vectors. If this "core" corpus is small, the collection of documents can be expanded with more general corpora that will provide additional evidences on the co-distribution of words and thereby mitigate the data scarcity. An alternative would be to use an embedding model pre-trained on large corpora (such as BERT) and fine tune it with the corpus of documents to be protected.

Since we aim at protecting entities and semantic relationships occur at a conceptual level (rather than at a word level), we introduce a pre-processing step to create a meaningful vocabulary of concepts (rather than isolated words) for the word embedding model.

Specifically, concepts and entities are referred to in a text via noun phrases. For example, the noun phrase "New York Times" refers to a sole specific entity that is completely different from the individual meaning of its words "New", "York" and "Times". To properly evaluate disclosure risks, we need the vector representation of the *concepts* referred to by the text (*e.g.*, "New York Times"), rather than the representations of isolated words. For this purpose, in the pre-processing step we extract the noun phrases (or n-grams) and feed them as atomic elements to the word embedding model for training.

As shown in Figure 1, the pre-processing step consists of a pipeline of syntactic analyses: tokenization, part-of-speech tagging and chunking [34]. As a result, noun, verb and prepositional phrases are obtained. Also, to minimize the lexical variability of the noun phrases, stop words are removed during the tokenization step; in this way, the occurrences of n-grams like "the New York Times" and "New York Times" will contribute to the same vocabulary entry/word vector.

In addition to improving the characterization of the entities appearing in the documents, this pre-processing helps reduce the size of the vocabulary and, therefore, the training runtime.

Our method is not tied to a particular embedding model: it only needs accurate and exhaustive vector representations of all the n-grams appearing in the documents to be protected. In the sequel we illustrate on word2vec [31] the training process of a word embedding model tailored to our needs, even though, as we show in the evaluation section, other embedding models can be employed. Word2vec can be trained either to predict the current word from a window of neighboring words (continuous bag-of-words model) or to predict neighboring words based on the current word (skip-gram model). To this end, the neural network uses a collection of sentences as training data, and builds a vocabulary with the words appearing in the collection. The weights of the hidden layer of neurons that results from training the neural network for each word in the vocabulary are used as the vector associated with that word. In this way, the number of neurons in the hidden layer (which can be configured), corresponds to the dimensionality of the vectors.

Regarding the learning model, the skip-gram model yields more accurate vectors [31]. More specifically, it uses as input a binary vector in which each position corresponds to a word w_i in the vocabulary V. The position of the current word (w_c) is set to '1', whereas the remaining positions are set to '0'. The output layer is a Softmax classifier with as many neurons as words in the vocabulary, where the i-th neuron provides the probability that the word at a randomly chosen nearby position of the current word w_c is w_i. The neural network is trained with nearby word pairs from the input collection of sentences. Context windows are employed to restrict the neighborhood of words that are considered to be nearby and to build the training samples. Once the training is complete for w_c, the weights of the hidden layer of neurons —which embed the tendency to co-occur between w_c and all the other words in the vocabulary— are employed as the vector representation of w_c.

The training of the skip-gram model depends on several configuration parameters. In what follows we discuss such parameters and argue which values are appropriate in the context of document protection.

As said above, the skip-gram model predicts the probability that words in the vocabulary appear in the neighborhood of the input word. To this end, it uses training samples of word pairs that co-occur within a context. This context (and, therefore, the co-occurring word pairs) can be restricted according to a *window size*. The window size is usually set to encompass complete sentences, say between 5 and 10 words each, because words appearing in the same sentence are assumed to be closely related. Larger window sizes require more iterations, because more word pairs are evaluated during the learning process; as a matter of fact, doubling the window size increases the training runtime by around 50%. We also set the window size to include sentences but considering that our linguistic units are n-grams rather than isolated words.

Another relevant parameter is the *dimensionality of vectors*. In principle, the greater the dimensionality, the more accurate the results, because the adaptability of the model is proportional to the vector size. However, since the dimensionality is equal to the number of neurons in the hidden layer of the network, a greater dimensionality significantly increases the training runtime. Again, doubling the size of the vectors implies increasing the runtime by around 50%. Even though there is no fixed rule to tune the dimensionality, a value 300 is suggested in [31] because larger values do not significantly improve accuracy.

Finally, it is possible to set a minimum number of appearances as a *cutting threshold* below which words in the input collection of documents will not be used for training. Since word embedding is usually employed to guide general semantic similarity assessments, it makes sense to discard words that occur too rarely because the evidences of co-occurrence they provide are too weak to derive robust statistics. Moreover, filtering out outliers significantly reduces the vocabulary size and, therefore, the training runtime. However, in the context of document protection rare words (such as names or particular addresses, which may appear only once together with the entity they refer to) are usually those that entail the greatest risk because they often refer to very specific (quasi-)identifying information [42]. For this reason, we do not use any cutting threshold for rare words. For such words, the model may learn a strongly biased relationship w.r.t. the entity to be protected, which is the only one they co-occur with in the training data. This is however beneficial from the point of view of privacy protection, because in this way these rare words will be

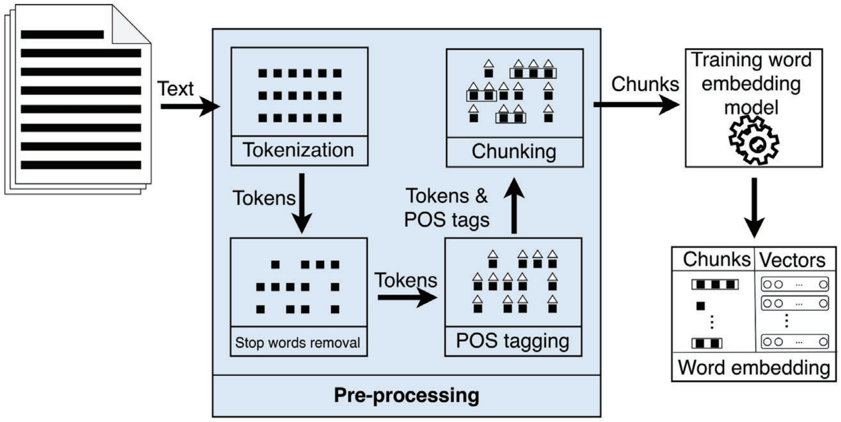

Fig. 1: Overview of the training phase

characterized for sure as quasi-identifying terms.

Training a word embedding model on a large collection of documents can be costly. Nonetheless, once trained, the model can be efficiently reused to protect any number of documents as long as their content is covered by the vocabulary of the trained model. Also, in the event that a new document to be protected contains terms that are not in the vocabulary, the previously trained model can be efficiently updated (without re-initializing the training) with new vocabulary entries via vocabulary expansion techniques [25].

3.2.2 Detecting Quasi-identifying Terms

Once the model is built, we obtain a vector representation of each phrase in the input collection of documents. If two n-grams had similar contexts in the training data (and, therefore, are semantically related [39]) they will also have similar vectors. The standard way of measuring the similarity between vectors is the cosine similarity. We employ this similarity to assess how disclosive/similar are the terms in a document w.r.t. the entity to be protected and, in this way, we detect those terms that act as quasi-identifiers of that entity.

The second phase of our method is depicted in Figure 2. Given a document D and a particular entity e whose privacy is to be protected (where e can be an identity or a confidential value), we iteratively evaluate how disclosive about e each phrase p_i in D is. We do this by measuring the cosine similarity between the vector representations of p_i and e, which we denote by $sim(v(p_i), v(e))$. Prior to that, we pre-process D as described in Section 3.2.1, so that the contents of D are evaluated at a conceptual level rather than at a word level. If the similarity of a certain p_i is above a threshold t, then p_i is deemed a quasi-identifier and will undergo masking in the third phase. Thus, t defines the maximum level of tolerated disclosure for the (masked) terms appearing in the protected output, and it allows balancing the trade-off between disclosure protection and utility preservation. As it happens with other generalization-based methods [6], [11], [14], which also rely on privacy/utility thresholds, the specific value of t should be set according to the needs of the application scenario: higher values for better protection (and less utility) or lower values for better utility (and less protection).

3.2.3 Masking Quasi-identifying Terms

As discussed in Section 2, different strategies can be employed to mask quasi-identifiers, of which the most common are suppression and generalization. The former strategy is straightforward and is usually employed in document redaction [43]. On the other hand, term generalization, which consists in replacing specific terms by less detailed generalizations, does a better job at preserving the semantics and the readability of the protected document. Since by definition generalizations encompass a subset of the semantics of their respective specializations, generalization-based replacements preserve a subset of the original document semantics.

Generalizing requires detailed taxonomies from which suitable generalizations of disclosive terms can be obtained. Taxonomies suitable for non-specialized text can be obtained from general-purpose ontologies, such as WordNet [18] or YAGO [52]. More specifically, WordNet models the semantic relationships between 175,979 concepts, which are taxonomically organized under the common abstraction "entity". YAGO enriches WordNet's taxonomy by adding Wikipedia categories and articles; as a result, YAGO includes more than 10 million entities. These knowledge bases can be expected to cover most of the entities appearing in text. For specialized documents such as medical records, domain-specific knowledge bases can be used; for example, SNOMED-CT [49] models more than 311,000 clinical terms within several taxonomies.

Masking quasi-identifying terms is performed as follows. For each quasi-identifying phrase s_i detected in the second phase, we obtain an ordered set of generalizations $G(s_i)$ from an ontology by matching s_i to concept labels in the ontology. If s_i matches more than one concept (due to its being polysemic), we map it to its most probable sense/concept, based on the probability of occurrence available in the sources we use for generalization. If s_i is not found in the ontology, we look for simpler forms of the noun phrase by iteratively removing adjectives and nouns starting with the leftmost ones (*e.g.*, "metastatic breast cancer" → "breast cancer" → "cancer"). These simpler forms of s_i are also added in the first positions of $G(s_i)$ because they are actual generalizations of s_i. In this way, $G(s_i)$ contains generalizations of s_i ordered from most specific to most general. If s_i is a very specific

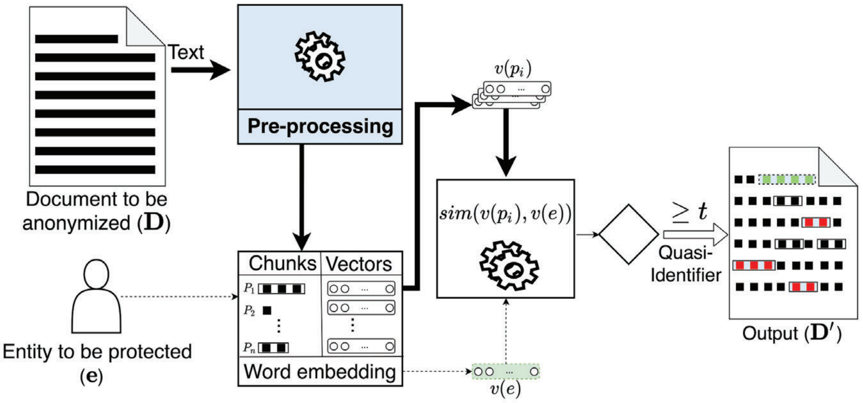

Fig. 2: Overview of the detection phase

entity, such as the name of an individual, we may not find it in any ontology. In this case, we use the most abstract concept in the ontology (*e.g.*, "entity") as its generalization.

As shown in Figure 3, the most suitable generalization g_i to mask s_i is the most specific generalization in $G(s_i)$ such that $sim(v(g_i), v(e)) < t$, that is, the first generalization in $G(s_i)$ that brings the disclosure on e below threshold t. To calculate $sim(v(g_i), v(e))$ we also need the vectors corresponding to all the generalizations of all s_i. Since g_i are generalizations of s_i, they are likely to have already appeared in the input collection of documents, in which case $v(g_i)$ has already been calculated. Otherwise, we need to update the model by feeding new documents that contain the missing g_i. Training data could be Wikipedia articles covering g_i, which are already associated with concepts in ontologies such as YAGO, and are a common training source of general-purpose embedding models [9]. As discussed in Section 3.2.1, to efficiently re-train the model with new samples (and vocabulary elements) we can use vocabulary expansion techniques. If the amount of documents and entities to be protected is large, a more efficient approach would be to first train a complete model covering all the entities contained in the ontology by using, *e.g.*, their corresponding Wikipedia articles as training data, and second, to expand the model by considering the specific contents of the documents to be protected. In this way, we ensure that all possible generalizations are already covered by the model when we reach the masking phase.

4 Evaluation

In this section we report a performance evaluation of our approach from three perspectives: (i) the accuracy of the detection of quasi-identifying terms, (ii) the utility preserved by the protected document after masking such terms, and (iii) the effectiveness of the protection against a simulated re-identification attack. We have evaluated our method under different conditions and we have compared our results against several tools based on named entity recognition.

4.1 Detection Phase

Our evaluation considered a scenario similar to that used in related works on document protection [12], [42], [43]. In these works, the evaluation data consist of a set of Wikipedia articles describing real-world entities of different domains. Our goal was to protect each article so that the outcome did not unequivocally disclose the entity described by the article. To obtain the ground truth, we manually examined the contents of the articles to identify the terms that might disclose the described entity. Wikipedia articles were used because of their high informativeness and tight discourses, which constitute a challenging scenario for document protection.

More specifically, we used a collection of English Wikipedia articles corresponding to movie actors from several countries. First, we collected the abstracts of 19,000 articles under the "20th century actors" Wikipedia category. These were used to train the word embedding model as detailed in Section 3.2.1. The model was built using word2vec [32]. Training was configured with the parameters discussed in Section 3.2.1: window size 10, vector dimension 300 and no filtering of rare words.

As an evaluation test bed, we randomly picked 50 summaries from the collection and we tagged them manually to identify words and n-grams that might disclose the actor's identity. We used the following annotation guidelines, which are inspired by how (quasi-)identifying attributes are selected in structured databases [23]:

- *Identifiers*: any information that can directly and unequivocally identify an individual. This includes the actor's name and also direct family members such as father, mother, brothers, children, husband/wife, etc. We also considered the movie characters' name he/she have played.
- *Quasi-identifiers*: publicly available information that, in isolation, does not identify the individual but whose combination may. There is an unbounded number of information types that may act as quasi-identifiers, but they mainly boil down to demographic and spatiotemporal attributes such as age, date of birth, place of living, received awards, names and dates of the movies he/she has started, etc.

The annotation was independently carried out by the three authors of this work. A final annotation was thereafter agreed upon via majority voting. The inter-annotator agreement among the three of us was Fleiss' kappa = 0.869, which

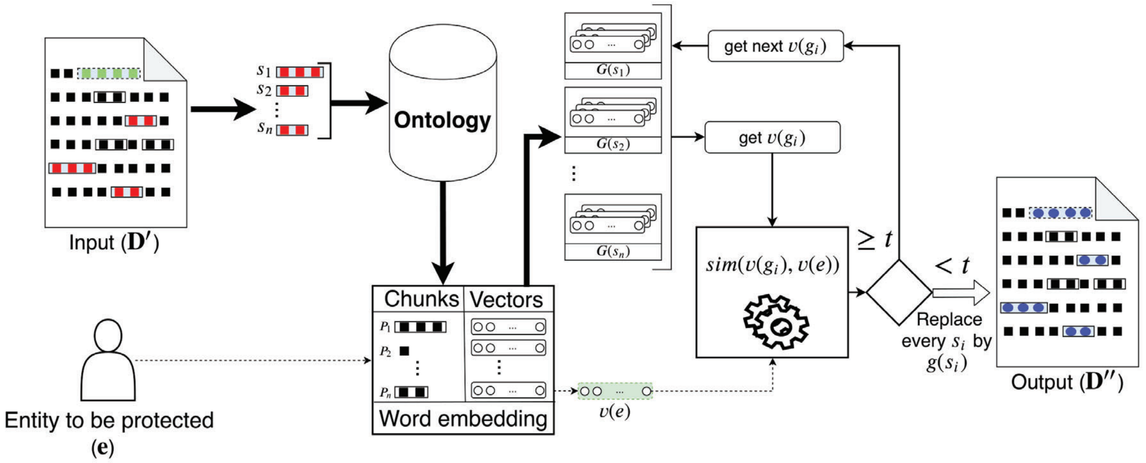

Fig. 3: Overview of the masking phase

shows a very strong agreement. As a result of the annotation, 2,655 words or around 30% of the content were tagged.

The evaluation metrics we employed were the standard precision, recall and F_1-score measures, which we next summarize. Precision is defined as

$$Precision = \frac{\#detected\ tagged\ terms}{\#detected\ terms},$$

where $detected\ terms$ is the set of terms detected as quasi-identifiers through the process detailed in Section 3.2.2 and $detected\ tagged\ terms$ is the subset of terms detected as quasi-identifiers that contain one or more tagged words. The higher the precision, the lower the number of false positives, that is, of over-protected terms. A high precision implies that the document's semantics and readability, that is, its utility, are better preserved by the protection process. Regarding recall, it is defined as

$$Recall = \frac{\#detected\ tagged\ terms}{\#tagged\ terms},$$

where $tagged\ terms$ is the set of terms manually tagged as quasi-identifiers. The higher the recall, the more robust the protection, because a greater amount of (quasi-)identifiers have been correctly detected. Finally, the F_1-score is defined as

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall},$$

which corresponds to the harmonic mean of precision and recall and can be viewed as a performance summary of the detection phase when the same weight is given to precision and recall. Notice that, even though a high precision is always positive, a high recall is usually more desirable because undetected quasi-identifiers may render the protection useless.

We empirically set the similarity threshold t employed to detect quasi-identifiers so that the F_1-score was maximized on average across the evaluated documents. The selected value was $t = 0.25$. Notice that, rather than being a hyperparameter to be optimized, the threshold t is a privacy requirement, *i.e.*, it allows tailoring the privacy/utility trade-off, and its value can be set by the user according to his/her protection needs. We show in Figure 4 how the threshold influences precision/recall/F_1 for values within the $[0.01, \ldots, 1]$ range. We can see that different values yield different balances between protection (recall) and utility preservation (precision), with $t = 0.25$ offering the best balance.

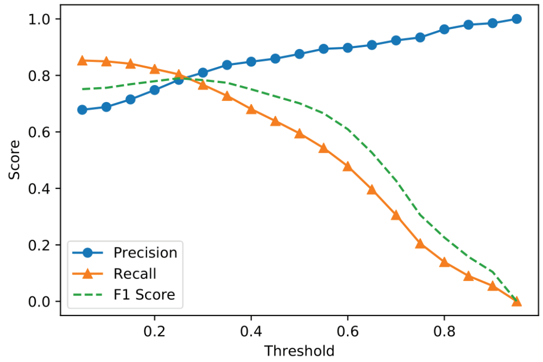

Fig. 4: Influence of the value of the similarity threshold t

We evaluated two versions of our method: the first one included the pre-processing detailed in Section 3.2.1 whereas the second one did not. In the first version the model vocabulary consisted of 651,835 n-grams, whereas in the second version it comprised 1,084,189 individual words. Model learning took 177 seconds with the first version and 232 seconds with the second version, in both cases on an AMD Athlon X4 860K CPU with 24GB RAM. Notice that document pre-processing is the only phase of our method that is language-dependent. Therefore, by measuring the influence of the linguistic pre-processing on the results, we were able to quantify the benefits brought by this additional analysis and the penalty incurred if the linguistic tools required to analyze a (minority) language were not available.

We then compared the evaluation figures obtained by our method against those achieved by several NER-based tools. In addition of NER being the most common approach to document protection, it is also the only method among those discussed in Section 2 that can compare with our approach in terms of practicality for real-world tasks. In particular we used the Stanford Named Entity Recognizer software [28], which provides 3 pre-trained NER models for English (NER3, NER4 and NER7), and Microsoft Presidio, which tailors NER towards privacy protection:

- NER3: detects and categorizes named entities of ORGANIZATION, LOCATION and PERSON types.
- NER4: detects and categorizes named entities of ORGANIZATION, LOCATION, PERSON and MISC types.
- NER7: detects and categorizes named entities of LOCATION, ORGANIZATION, DATE, MONEY, PERSON, PERCENT and TIME types.
- Presidio: detects and categorizes named entities of CREDIT_CARD, CRYPTO, DATE_TIME, DOMAIN_NAME, EMAIL_ADDRESS, IBAN_CODE, IP_ADDRESS, LOCATION, PERSON, NRP, PHONE_NUMBER, UK_NHS, US_BANK_NUMBER, US_DRIVER_LICENSE, US_ITIN, US_PASSPORT and US_SSN types.

Table 1 reports the evaluation figures of the different methods on average for the 50 documents under consideration. It is clear that our method improves on the NER-based approach very significantly, regardless of the NER model in use. In particular, the recall is more than doubled, which results in a much higher F_1-score. This illustrates the main limitation of NER-based methods: named entities are not the sole source of disclosure. This limitation tends to yield under-protected documents in which, for example, identities may be disclosed by correlating several non-protected facts or personal features that do not fall into the predefined types of named entities. By comparing the last column of Table 1 with Figure 4, we also see that our method provides significantly better F_1 scores than NER tools for a wide range of threshold values (those below 0.5). This shows that the user enjoys some freedom to tailor the threshold to his/her needs, while still getting a better protection-utility balance than with NER-based methods.

Regarding the differences among the three NER models, we see that NER3 produces the worst recall because it has been trained to detect the least number of entity types. NER7 and Presidio improve on the results of NER3, mainly because they can detect dates such as birthdates, which are quite common in biographies. Finally, whereas NER3, NER7 and Presidio have been trained with specific named entity types, NER4 adds the MISC type, which encompasses a variety of named entities such as nationalities. No significant differences in precision are visible across NER tools, regardless of the different models they use to detect NEs, *i.e.*, CRF for Standford and BERT-based NER for Presidio. From the privacy point of view, the low recall resulting from the limited amount of supported NE types has a much greater influence than precision.

Disabling the pre-processing in our method decreases the recall from 81.24% to 59.79%. Even though this penalty is large, the decreased recall is still significantly higher than the recall of the best NER model. On the one hand, this illustrates the benefits of analyzing the content of documents at a conceptual level, rather than at a word level. On the other hand, it can be seen that the results of our approach with a language-agnostic implementation (*i.e.*, without language-dependent tools) are still significantly better than those of NER-based methods, which nonetheless require language-specific tagged training data.

TABLE 1: Average precision, recall and F_1-score for the 50 evaluated documents

Method	*Precision*	*Recall*	F_1
NER3	96.09%	19.59%	32.07%
NER4	97.59%	34.25%	49.72%
NER7	**98.32%**	27.89%	42.77%
Presidio	98.06%	27.07%	41.12%
Our method	82.69%	**81.24%**	**81.66%**
Our method (no pre-process)	83.48%	59.79%	69.00%

The behavior of the different methods is illustrated in Table 2, which contains an extract of the input text of one of the evaluated documents and compares the manual tagging with the entities detected by the different approaches. We can see that NER-based methods failed to detect pieces of information that are relevant to re-identify the actor, such as her/his birth date (for NER3 and NER4) or the title of the movies or TV series she/he appeared in. NER7 is particularly worrying, because it missed the actor's name, which is a direct identifier. In contrast, our approach only missed the actor's profession (due to its being very general), and only incurred over-protection for the term "the action drama".

In fact, it takes more than providing good average results for a method to be useful: a good method has to yield good enough results in all cases. Table 3 reports the coefficient of variation (a measure of dispersion computed as the ratio of the standard deviation to the mean) of the results given in Table 1. We can see that our approach provides the most consistent results, with a variation of the F_1-score just 0.31%.

Precision is the only metric for which the NER-based approach achieved better figures. Indeed, NER has an inherently high accuracy in a pure NER task. Moreover, the evaluation scenario we consider is quite favorable to NER because most of the text in each document is highly related to the individual to be protected (the biographee). Therefore, if a named entity appeared in the text and was properly identified by the NER method, then this named entity was very likely to refer to the biographee and, therefore, to be disclosive. In a less favorable scenario, in which the content of a document could refer to different people, the precision of the NER-based approach would significantly decrease, because not all the named entities in the document would refer to the individual to be protected. We simulated this setting by putting together the biographies of two related actors (both American and acting in the same TV series) and manually tagging only the terms that may be disclosive on one of them. In this case, the system had to detect not only those terms that exclusively related to the actor to be protected, but also the information he or she had in common with the other actor also referred to in the text. The results of this experiment are reported in Table 4.

As expected, the precision of the NER-based methods is significantly lower in this two-actor setting, even though we see relevant differences among the different NER models. The problem was not only to detect the NEs, but to distinguish which actor an NE referred to. Some significant false positives

TABLE 2: Output samples for each method

Manual annotation	Thomas Cruise Mapother IV (born July 3, 1962) is an American actor and producer. He started his career at age 19 in the film Endless Love (1981), before making his breakthrough in the comedy Risky Business (1983) and receiving widespread attention for starring in the action drama Top Gun (1986) as Lieutenant Pete "Maverick" Mitchell.
NER3	Thomas Cruise Mapother IV (born July 3, 1962) is an American actor and producer. He started his career at age 19 in the film Endless Love (1981), before making his breakthrough in the comedy Risky Business (1983) and receiving widespread attention for starring in the action drama Top Gun (1986) as Lieutenant Pete "Maverick" Mitchell.
NER4	Thomas Cruise Mapother IV (born July 3, 1962) is an American actor and producer. He started his career at age 19 in the film Endless Love (1981), before making his breakthrough in the comedy Risky Business (1983) and receiving widespread attention for starring in the action drama Top Gun (1986) as Lieutenant Pete "Maverick" Mitchell.
NER7	Thomas Cruise Mapother IV (born July 3, 1962) is an American actor and producer. He started his career at age 19 in the film Endless Love (1981), before making his breakthrough in the comedy Risky Business (1983) and receiving widespread attention for starring in the action drama Top Gun (1986) as Lieutenant Pete "Maverick" Mitchell.
Presidio	Thomas Cruise Mapother IV (born July 3, 1962) is an American actor and producer. He started his career at age 19 in the film Endless Love (1981), before making his breakthrough in the comedy Risky Business (1983) and receiving widespread attention for starring in the action drama Top Gun (1986) as Lieutenant Pete "Maverick" Mitchell.
Our method	Thomas Cruise Mapother IV (born July 3, 1962) is an American actor and producer. He started his career at age 19 in the film Endless Love (1981), before making his breakthrough in the comedy Risky Business (1983) and receiving widespread attention for starring in the action drama Top Gun (1986) as Lieutenant Pete "Maverick" Mitchell.

TABLE 3: Average coefficients of variation (CV) for precision, recall and F_1-score

Method	*Precision CV*	*Recall CV*	*F_1 CV*
NER3	0.48%	2.14%	2.41%
NER4	0.16%	3.67%	2.97%
NER7	**0.09%**	2.65%	2.34%
Presidio	0.12%	4.75%	5.02%
Our method	0.52%	**0.67%**	**0.31%**

TABLE 4: Precision, recall and F_1-score for a document referring to two different individuals

Method	*Precision*	*Recall*	*F_1*
NER3	55.55%	22.72%	32.25%
NER4	**77.27%**	59.09%	66.96%
NER7	60.0%	27.27%	37.5%
Presidio	66.67%	38.1%	48.48%
Our method	68.0%	**81.81%**	**74.27%**

of NER tools involved tagging the birth place and birth date of the actor *not* to be protected; this was a mistake that our method avoided. Although the false positive rate of our method also increased with respect to the single-actor setting, the increase was smaller than for NER-based methods; besides, the recall of our method stayed at the same level as in the single-actor setting.

So far, we have examined the performance of our method on word2vec and with an excellent training data set that perfectly matches the contents of the evaluated documents. However, gathering large and suitable training data may be difficult in some domains. On the other hand, our method is not tied to a particular embedding model and may benefit from advances in embedding techniques. To assess the generality of our approach, we also experimented with the following word embeddings models trained on general-purpose data:

- *Pre-trained word2vec* [32]: an off-the-shelf word2vec model trained on the Google News data set. The model has a vocabulary of 3 million words/terms.
- *FastText* [9]: a library for word embedding learning created by Facebook's AI Research lab (see Section 3.1 for more details). Two pre-trained models were considered: the first model (*wiki1*) has a vocabulary of 2 million words/terms trained on the Common Crawl data set, which is an archive of web data collected since 2011; the second model (*wiki2*), has a vocabulary of 1 million words/terms trained on the 2017 Wikipedia snapshot, the UMBC webbase corpus and the statmt.org news data set.
- *BERT (base-cased)* [15]: a BERT model with 12 encoders with 12 bidirectional self-attention heads trained from data extracted from the BookCorpus with 800M words and the English Wikipedia with 2,500M words.

When the training data do not perfectly match the contents of the document to which the model is applied, the document may contain out of vocabulary (OOV) terms. For the models trained with fastText this does not occur because it approximates OOV vectors from subword information. However, since word2vec does not do this, many of the complex n-grams we extracted from the documents to be evaluated were not found in the model's vocabulary. For the Google News model to provide usable results, we had to disable pre-processing so that the content of the document was evaluated at a word level. The evaluation figures obtained with the pre-trained models are reported in Table 5.

It is interesting to see that the models trained with fastText and BERT produced results comparable to those obtained with our domain-specific training corpora. This shows that in the absence of such domain-specific corpora, large general-purpose corpora and pre-trained models may be employed with reasonably good results. However, when the pre-processing applied to pre-train the model does not match the pre-processing used to evaluate new documents

TABLE 5: Evaluation figures with several pre-trained word embedding models

Model	*Precision*	*Recall*	F_1
FastText (wiki1)	82.38%	71.84%	75.93%
FastText (wiki2)	83.06%	71.85%	76.20%
Word2Vec (Google News)	68.58%	24.80%	35.58%
BERT (base-cased)	81.84%	72.31%	75.95%

and OOV terms are not handled properly, results are much worse, as it was the case for the Google News model. Recall was especially bad because many of the n-grams that ought to have been detected as quasi-identifiers were either not found in the vocabulary or were only partially detected, which we also counted as a miss.

4.2 Masking Phase

In this section we report on the performance of the masking strategy presented in Section 3.2.3. We measured the relative utility preserved by the protected document after masking via ontology-based generalization the quasi-identifiers detected in the previous phase. Generalizations for quasi-identifiers were obtained from WordNet and YAGO. The vectors of such generalizations were computed by re-training the model with the Wikipedia articles corresponding to those generalizations.

Similarly to related works [44], we measured the relative utility of the protected document (D'') as the aggregation of the semantics it conveys w.r.t. the semantics of the original document (D). This yields

$$Utility\_preservation(D'') = \frac{Semantics(D'')}{Semantics(D)} \cdot 100,$$

where $Semantics(D)$ is the sum of the information content $IC(p_i)$ of each phrase p_i in D, that is,

$$Semantics(D) = \sum_{i=1}^{n} IC(p_i),$$

with n being the number of phrases in D.

In information-theoretic terms, $IC(p_i)$ is the inverse logarithm of the probability of occurrence of p_i:

$$IC(p_i) = -log_2 \Pr(p_i).$$

To obtain representative probabilities of occurrence, we queried p_i in the Bing search engine and divided the number of results it provides by the total number of resources indexed by the search engine, as done in [44].

The utility metric we use captures the fact that the generalizations used for masking carry less information than their respective specializations. The information content is a metric commonly used to quantify the semantic content of terms in computational linguistics [38]. Moreover, in the literature on data privacy [23], utility preservation is usually measured as a function of the information loss incurred by masking, which is exactly what our utility metric does. We focus on the total information content lost as a result of the replacements rather on the number of such replacements.

Table 6 depicts the average relative utility preserved by different methods and masking strategies for the 50 evaluated documents. We compared our method against the NER approaches discussed in the previous section when replacing the detected named entities by their types (*e.g.*, "Tom Cruise" → PERSON). We also report the relative utility that remained when quasi-identifiers (in our case) and named entities (for NER-based methods) were suppressed, as usually done in document redaction [43].

TABLE 6: Average relative utility preserved by different methods and masking strategies

Method	*Suppression*	*Generalization*	*Avg. masked terms*
NER3	**66.98%**	**85.44%**	33.8
NER4	39.70%	74.73%	64.08
NER7	54.00%	81.29%	54.16
Presidio	54.04%	81.00%	61.12
Our Method	48.48%	85.01%	**86.04**

As expected, plain suppression produced protected outcomes that retained significantly less utility than generalization. Add to this that blacking out pieces of text hampers document readability and makes potential attackers aware of the document sensitivity [8]. Generalization, either via ontologies or via named entity types, preserved much more utility. For NER models, utility figures were inversely proportional to the number of masked terms (shown in the last column). In contrast, our approach yielded the second best utility value while masking the largest number of terms (resulting from the best recall in the detection phase). The good utility was due to the use of fine-grained ontological generalizations rather than coarse NE types. Thus, our method achieved the best balance between privacy protection and utility preservation.

4.3 Protection Against Re-identification

So far we have evaluated detection and masking in isolation. To measure the practical effectiveness of protection as a whole, we implemented a re-identification attack that is inspired by the evaluation framework proposed in works like [19] for authorship attribution. The general idea of this experiment is to check the ability of a machine learning classifier to correctly predict the entity from the protected output of each method.

Specifically, we took the 50 articles we had manually annotated and we fine-tuned the BERT base-cased model to predict the actor's name by training it on the post-summary text, that is, all of the article's text except the part we manually annotated. We split the post-summary text into sentences, each one labeled with the name of the actor. We used 80% of the sentences to train the model and the remaining 20% to validate it. Then we tested the classifier on the summary text, which is the part that we manually annotated and that we protected. Predictions were evaluated by checking whether the majority-predicted class of the sentences in the summary matched the actual actor. The classifier was tested on the original unchanged summaries, the manually annotated summaries (by just replacing the tagged text by the label SENSITIVE) and the masked outputs of the different protection methods. The percentage of correct predictions is reported in Table 7. Due to the non-deterministic behavior of the BERT model in

TABLE 7: Percentage of correct predictions for each method

Input	*Correct predictions*
Original summary	84.67%
Manual annotation	2.00%
NER3	18.00%
NER4	16.00%
NER7	37.33%
Presidio	18.67%
Our Method	10.00%

tensorflow, which sightly varies for every run, we report the average results of three runs.

First of all, it is important to highlight that this setting is very favorable for re-identification. On the one hand, the number of individuals/classes to be predicted is very limited in comparison with the size of the population of personal data sets (accounting for thousands or millions of individuals). On the other hand, the text used for prediction (summary) bears a lot of similarities to the training data, not only regarding content, but also regarding the linguistic structure of the sentences. As a matter of fact, sentences in the summary also appear quite frequently in the post-summary text, and this gives an 'unfair' advantage to the classifier. Despite all the above, we see in Table 7 that the prediction accuracy for manual annotation was at the level of random guess (2%, that is, $1/50$). Yet the protection achieved by manual annotation came at a high utility cost, because the masking in this case was equivalent to text suppression and suppression was shown in Table 6 to significantly damage the utility of the document. Table 7 also shows that our method is the one that offers the best protection, closest to the protection level offered by manual annotation and with much less utility loss (due to the use of ontological generalization).

The results in Table 7 show some discrepancy with respect to the recall-based protection reported in Table 1 for some NER models, especially NER7. Even though NER7 yielded a higher recall than NER3 due to the former considering a larger variety of NE types, its protection against re-identification was less effective, mainly because NER7 failed to detect some family names that NER3 did not miss, as we mentioned above. This illustrates that recall figures do not give a complete view on the robustness of protection, because the nature of the terms missed by a method (*e.g.*, highly disclosive direct identifiers such as family names or less risky circumstantial quasi-identifiers such as the year an event happened) may be more influential on the success of re-identification attacks than the number of identified terms.

5 Application Scenarios

The approach we present in this paper is remarkably versatile and unconstrained. In particular, it does not require manually tagged data, it works reasonably well with general-purpose pre-trained models and, except for the optional pre-processing, it is language-agnostic. As a result, our method can be immediately applied to a variety of real-word scenarios.

The most natural application of text protection is document declassification, which consists in releasing documents that used to be classified as confidential. Declassification is oftentimes motivated by transparency principles and open data initiatives. To make transparency compatible with data protection and other interests at stake, parts of the declassified documents that may refer to non-public individuals, facts or places need to be sanitized by redacting (blacking out or deleting) them. Redaction is also employed for selective disclosure of information. For example, when a document is subpoenaed in a court case, information not relevant to the case is often redacted. Similarly, US legislations on the privacy of medical data mandate hospitals to redact all direct or indirect references to sensitive diseases (such as sexually transmitted diseases or AIDS) before releasing patient records to insurance companies or in response to worker's compensation or motor vehicle accident claims [8]. As discussed in Section 2, redaction has traditionally been performed manually by following certain rules or guidelines [35]. However, manual approaches are time-consuming [17] and error-prone, and they usually require the coordinated effort of several human experts [8]. Our method perfectly fits the needs of document redaction: given a set of entities to be protected (identities, locations or confidential values such as sensitive diseases), our technique can be iteratively applied to each entity in a given document so that any references, either direct or indirect, to those entities are detected and subsequently redacted.

In a different context, the well-known Snowden and Wikileaks scandals have made companies more aware of the damage that may be caused by insiders who gradually gain access to more and more confidential data. To mitigate this threat, companies have started to implement risk management policies, whereby the contents of corporate files are characterized according to their risk, and accounting is enforced on employees by continuously monitoring their accesses to such files. Then, metrics such as *misuseability scores* [21] can be developed to quantify the harm that might be inflicted by an employee in a hypothetical data leakage as a function of the accumulated sensitive data he or she has accessed. These metrics enable early detection and prevention of data leakage or misuse by insiders, for example, by implementing dynamic access control policies to decide whether or not access to new content should be granted to specific employees, or by detecting individuals with unusually high scores. A variety of commercial software packages are available to enforce risk assessment on corporate files, such as the aforementioned Amazon's Macie [4], Google's DLP [13] or Symantec's Data Loss Prevention [53]. However, all those packages characterize risk based on the (limited set of) named entity types they can detect by means of regular expressions and pre-trained classifiers. Thus, they suffer from the limitations discussed in Section 2. In this respect, as shown in Section 4, our approach can offer a much more accurate risk characterization, which can also be tailored to the specific privacy requirements of the organization.

A similar approach can also be employed to measure the exposure level of users of social networks and therefore their privacy risks. Proposals in the literature compute privacy risk scores of social network users as the sum of attributes disclosed by their profiles [27], [51]. However, messages posted by users provide much more detailed and up-to-date information on the users' preferences or demography than static attributes, thereby entailing higher risk [47]. Our method can be applied (trained) on the users' data and be enforced on the topics that current regulations (such as GDPR) regard as sensitive, such as religion, sexuality or ethnicity. As a consequence, the user can be made aware of the level of exposure his or her publications entail on such sensitive topics and by that means he or she can make informed decisions on whether to publish certain data. User awareness and empowerment regarding privacy are in fact pillars of the modern outlook on privacy protection [46].

6 Conclusions and Future Work

We have presented an automatic method to protect text documents that leverages word embeddings to measure disclosure risk and masks disclosive terms via utility-preserving generalizations. Our approach is more general and, at the same time, more flexible than methods based on NER. On the one hand, we do not restrict the disclosure assessment to predefined entity types, because doing so typically incurs under-protection, as we have shown in our evaluation. On the other hand, our method drives protection according to privacy requirements focused on the entity or entities on whom information should not be disclosed by the sanitized text. This behavior is more similar to the way human experts tackle manual sanitization [8] and to the way privacy models enforce *ex-ante* privacy guarantees in structured databases [16]. As a result, the protection afforded by our method is consistent with the privacy requirements and, at the same time, more robust and utility-preserving than the protection of NER-based methods. Finally, even though our method relies on machine learning, it does not require tagged data and model building is language-agnostic. Therefore, no manual effort is required during the whole lifecycle of the protection process, which makes our method suitable for managing large amounts of textual data.

As future work, we plan to tailor contextual embedding models like BERT to our domain. As shown in the evaluation, pre-trained BERT was able to obtain results comparable to a word2vec model trained on domain-specific data. Hence, BERT trained on domain-specific data might offer even better results. Moreover, thanks to the contextual embeddings provided by BERT, language ambiguity will be minimized without requiring complex semantic disambiguation methods.

Acknowledgments

We acknowledge support from the European Commission (projects H2020-871042 "SoBigData++" and H2020-101006879 "MobiDataLab"), the Government of Catalonia (ICREA Acadèmia Prizes to J. Domingo-Ferrer and D. Sánchez, and grant 2017 SGR 705), the Spanish Government (projects RTI2018-095094-B-C21 "Consent" and TIN2016-80250-R "Sec-MCloud") and the Norwegian Research Council (project no. 308904 "CLEANUP"). We are with the UNESCO Chair in Data Privacy, but the views in this paper are our own and not necessarily shared by UNESCO.

References

[1] D. Abril, G. Navarro-Arribas, and V. Torra. On the declassification of confidential documents. In *Proceedings of Modeling Decisions for Artificial Intelligence, MDAI 2011*, pages 235–246, 2011.

[2] D. Acemoglu and A. Ozdaglar. Opinion dynamics and learning in social networks. *Dynamic Games and Applications*, 1(1):3–49, 2011.

[3] A. Akbik, T. Bergmann, and R. Vollgraf, "Pooled contextualized embeddings for named entity recognition," in *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2019, p. 724–728.

[4] *Amazon Macie - Amazon Web Services (AWS)*. https://aws.amazon.com/macie/. Last accessed: 24-Jan-2020.

[5] B. Anandan and C. Clifton. Significance of term relationships on anonymization. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, Vol. 03, pages 253–256. IEEE Computer Society, 2011.

[6] B. Anandan, C. Clifton, M. Jiang, W. Murugesan, P. Pastrana-Camacho, and L. Si. t-plausibility: Generalizing words to desensitize text. *Transactions on Data Privacy*, 5(3):505–534, 2012.

[7] M. Batet and D. Sánchez. Semantic disclosure control: semantics meets data privacy. *Online Information Review*, 42(3):290–303, 2018.

[8] E. Bier, R. Chow, P. Golle, T. H. King, and J. Staddon. The rules of redaction: Identify, protect, review (and repeat). *IEEE Security & Privacy*, 7(6):46–53, 2009.

[9] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[10] G. Boleda, "Distributional semantics and linguistic theory," *Annual Review of Linguistics*, vol. 6, pp. 213–234, 2020.

[11] T. Chakaravarthy, H. Gupta, P. Roy, and M. Mohania. Efficient techniques for document sanitization. In *Proceedings of the ACM Confernece on Information and Knowledge Management*, pages 843—-852, 2008.

[12] R. Chow, P. Golle, and J. Staddon. Detecting privacy leaks using corpus-based association rules. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 893–901. ACM, 2008.

[13] *Cloud data loss prevention*. https://cloud.google.com/dlp/. Last accessed: 24-Jan-2020.

[14] C. Cumby and R. Ghan. A machine learning based system for semi-automatically redacting documents. In *Proceedings of the 23rd Innovative Applications of Artificial Intelligence Conference*, pages 1628–1635, 2011.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[16] J. Domingo-Ferrer, D. Sánchez and J. Soria-Comas. *Database Anonymization: Privacy Models, Data Utility, and Microaggregation-Based Inter-Model Connections*, Morgan & Claypool, 2016.

[17] D. Dorr, W. Phillips, S. Phansalkar, S. Sims, and J. Hurdle. Assessing the difficulty and time cost of de-identification in clinical narratives. *Methods of Information in Medicine*, 45(3):246–252, 2006.

[18] C. Fellbaum (ed.). *WordNet: An Electronic Lexical Database*. MIT, 1998.

[19] N. Fernandes, M. Dras and A. McIver. Generalised Differential Privacy for Text Document Processing. In *Proc. of the 8th International Conference of Principles of Security and Trust (POST 2019)*, pages 123–148. Lecture Notes in Computer Science, Springer, 2019.

[20] O. Feyisetan, T. Diethe and T. Drake. Leveraging Hierarchical Representations for Preserving Privacy and Utility in Text. In *Proc. of 2019 IEEE International Conference on Data Mining (ICDM)*, pages 210–219. 2019.

[21] A. Harel, A. Shabtai, L. Rokach, and Y. Elovici. M-score: A misuseability weight measure. *IEEE Transactions on Dependable and Secure Computing*, 9(3):414–428, 2012.

[22] F. Hassan, D. Sánchez, J. Soria-Comas, and J. Domingo-Ferrer. Automatic anonymization of textual documents: Detecting sensitive information via word embeddings. In *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 358–365. IEEE, 2019.

[23] A. Hundepool, D. Domingo-Ferrer, L. Franconi, S. Giessing, E.S. Nordholt, K. Spicer, and P.P. de Wolf. *Statistical Disclosure Control*. Wiley, 2012.

[24] N. Jakob, S.H. Weber, M.C. Muller, and I. Gurevych. Beyond the stars: exploiting free-text user reviews to improve the accuracy of

movie recommendations. In *Proc. of the 1st Intl. CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion*, pages 57–64, 2009.

[25] R. Kiros, Y. Zhu, R.R. Salakhutdinov, Zemel R., R. Urtasun, A. Torralba, and S. Fidler. Skip-thought vectors. In *Proc. of Advances in Neural Information Processing Systems (NIPS 2015)*, 2015.

[26] Y. Li, T. Baldwin and T. Cohn. Towards Robust and Privacy-preserving Text Representations. In *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 25–30. 2018.

[27] K. Liu and E. Terzi. A framework for computing the privacy scores of users in online social networks. In *Proc. of the 9th IEEE International Conference on Data Mining (ICDE 2009)*, pages 288–297. IEEE Computer Society, 2009.

[28] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, 2014.

[29] S.M. Meystre, F.J. Friedlin, B.R. South, S. Shen, and M.H. Samore. Automatic de-identification of textual documents in the electronic health record: a review of recent research. *BMC Medical Research Methodology*, 10(70), 2010.

[30] Microsoft, "Presidio - data protection api," https://github.com/microsoft/presidio, 2019.

[31] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proc. of Advances in Neural Information Processing Systems (NIPS 2013)*, pages 3111–3119, 2013.

[32] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

[33] S.M. Mohammad and G. Hirst. Distributional measures of semantic distance: A survey. arXiv preprint arXiv:1203.1858, 2012.

[34] T. Morton, J. Kottmann, J. Baldridge, and G. Bierner. OpenNLP: A Java-based NLP toolkit. In *Proc. of the 10th Conf. of the European Chapter of the Association for Computational Linguistics (EACL 2005)*, 2005.

[35] National Security Agency. *Redacting with confidence: How to safely publish sanitized reports converted from Word to PDF*. Report #I333-015R-2005, Dec. 13, 2005.

[36] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[37] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[38] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. arXiv preprint cmp-lg/9511007, 1995.

[39] M. Sahlgren. The distributional hypothesis. *Italian Journal of Disability Studies*, 20:33–53, 2008.

[40] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(8):1010–1027, 2001.

[41] D. Sánchez, M. Batet, A. Valls, and K. Gibert. Ontology-driven web-based semantic similarity. *Journal of Intelligent Information Systems*, 35(3):383–413, 2010.

[42] D. Sánchez, M. Batet, and A. Viejo. Automatic general-purpose sanitization of textual documents. *IEEE Transactions on Information Forensics and Security*, 8(6):853–862, 2013.

[43] D. Sánchez, M. Batet, and A. Viejo. Minimizing the disclosure risk of semantic correlations in document sanitization. *Information Sciences*, 249:110–123, 2013.

[44] D. Sánchez and M. Batet. C-sanitized: A privacy model for document redaction and sanitization. *Journal of the Association for Information Science and Technology*, 67(1):148–163, 2016.

[45] D. Sánchez and M. Batet. Toward sensitive document release with privacy guarantees. *Engineering Applications of Artificial Intelligence*, 59:23–34, 2017.

[46] D. Sánchez and A. Viejo. Personalized privacy in open data sharing scenarios. *Online Information Review*, 41(3):298–310, 2017.

[47] D. Sánchez, J. Domingo-Ferrer, and S. Martínez. Co-utile disclosure of private data in social networks. *Information Sciences*, 441:50–65, 2019.

[48] C.C. Shilakes and J. Tylman. *Enterprise Information Portals*. Merrill Lynch Inc., New York, NY, 1998.

[49] K. Spackman. SNOMED-CT milestones: Endorsements are added to already-impressive standards credentials. *Healthcare Informatics*, 21:54–56, 2004.

[50] *spaCy: Industrial-Strength Natural Language Processing in Python*. Accessed April 27, 2021. https://spacy.io

[51] A. Srivastava and G. Geethakumari. Measuring privacy leaks in online social networks. In *2013 International Conference on Advances in Computing Communications and Informatics (ICACCI 2013)*, pages 2095-2100. IEEE, 2013.

[52] F.M. Suchanek, Kasneci G., and Weikum G. YAGO - a core of semantic knowledge unifiying WordNet and Wikipedia. In *Proceedings of the 16th International World Wide Web Conference (WWW 2007)*, pages 697–706, 2007.

[53] *Symantec data loss prevention*. https://www.symantec.com/products/dlp. Last accessed: 24-Jan-2020.

[54] The Economist. Data, data everywhere: A special report on managing information. *The Economist*, Feb. 27, 2010.

[55] United Nations. *Universal Declaration of Human Rights*, 1948 (article 12).

[56] A.F. Westin. *Privacy and Freedom*. Atheneum, New York, NY, 1967.

Fadi Hassan Fadi Hassan is a Ph.D. student at Universitat Rovira i Virgili. In 2012 he received a Bachelor's degree in Computer Science from Hodeidah University, Yemen. In 2017, he earned a Master's degree in Computer Security and Artificial Intelligence from Universitat Rovira i Virgili, Tarragona, Catalonia. His current research interests include machine learning, deep learning, natural language processing, and data privacy.

David Sánchez is an associate professor and an ICREA-Acadèmia Researcher at Universitat Rovira i Virgili, Tarragona, Catalonia. He received his Ph.D. in Computer Science from the Technical University of Catalonia (Barcelona) in 2008. He has participated in several National and European funded research projects and authored several papers and conference contributions. His research interests include data semantics, ontologies, machine learning and data privacy.

Josep Domingo-Ferrer (Fellow, IEEE) is a Distinguished Professor of Computer Science and an ICREA-Acadèmia Researcher at Universitat Rovira i Virgili, Tarragona, Catalonia, where he holds the UNESCO Chair in Data Privacy and leads CYBERCAT. He received the MSc and PhD degrees in Computer Science from the Autonomous University of Barcelona in 1988 and 1991, respectively. He also holds an MSc degree in Mathematics. His research interests are in data privacy, data security and cryptographic protocols. More information can be found at http://crises-deim.urv.cat/jdomingo