future internet

MDPI

Article

AI-Based Analysis of Policies and Images for Privacy-Conscious Content Sharing

Francesco Contu, Andrea Demontis, Stefano Dessì, Marco Muscas and Daniele Riboni *

Department of Mathematics and Computer Science, University of Cagliari, 09124 Cagliari, Italy;
fr.contu@outlook.com (F.C.); ad.demontis@gmail.com (A.D.); stefano.de.uni@gmail.com (S.D.);
muscas.marco@gmail.com (M.M.)
* Correspondence: riboni@unica.it

Abstract: Thanks to the popularity of personal mobile devices, more and more of the different types of private content, such as images and videos, are shared on social networking applications. While content sharing may be an effective practice to enhance social relationships, it is also a source of relevant privacy issues. Unfortunately, users find it difficult to understanding the terms and implications of the privacy policies of apps and services. Moreover, taking privacy decisions about content sharing on social networks is cumbersome and prone to errors that could determine privacy leaks. In this paper, we propose two techniques aimed at supporting the user in taking privacy choices about sharing personal content online. Our techniques are based on machine learning and natural language processing to analyze privacy policies, and on computer vision to assist the user in the privacy-conscious sharing of multimedia content. Experiments with real-world data show the potential of our solutions. We also present ongoing work on a system prototype and chatbot for natural language user assistance.

Keywords: privacy policy assistant; social networks; artificial intelligence

check for updates

Citation: Contu, F.; Demontis, A.; Dessì, S.; Muscas, M.; Riboni, D. AI-Based Analysis of Policies and Images for Privacy-Conscious Content Sharing. *Future Internet* **2021**, *13*, 139. https://doi.org/10.3390/fi13060139

Academic Editors: Salvatore Carta, Roberto Saia and Olaf Bergmann

Received: 30 April 2021
Accepted: 20 May 2021
Published: 21 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Personal mobile and IoT systems, including wearable sensors, smart objects, smartphones, and other devices that continuously acquire data, are a fundamental source of 'Big Data', and therefore an enabler of the data-based economy, as universally recognized. They are also a source of 'Small Data', defined as the portion of Big Data relating to an individual. Small Data are frequently shared by users in the form of digital content such as images or videos. The widespread nature of personal IoT ecosystems will therefore increase the amount of private data that is shared by orders of magnitude, as well as the entities from which the data can be acquired, aggregated and processed. This fact determines the relevant privacy issues that are of concern to people and authorities. For instance, the European legislation has recently enshrined the General Data Protection Regulation, which poses stringent rules on the protection of personal data and informed consent. In particular, it states that the consent request must be presented in an understandable and easily accessible form, using simple and clear language.

At present, however, the privacy policies of mobile and IoT devices and apps, including those for online social networking, are specified in text documents that are difficult to understand for ordinary users. In fact, recent studies have shown that when it is explained to mobile app users, what data are actually available to service providers and what the privacy implications are, users often feel surprised and deceived [1]. At the time of writing, there are no integrated and user-friendly mechanisms that allow users to understand the privacy implications of sharing their data in mobile and IoT ecosystems and to specify their privacy policies accordingly.

Current privacy protection solutions cannot be successfully applied to the mobile and IoT scenarios. Indeed, individuals cannot cope with the complexity of understanding each service or device's privacy preferences and the implications of its settings, which include what can be gleaned from analytics when merging information from multiple sources [2]. Numerous studies have proposed different tools and formalisms for expressing privacy preferences in various application contexts [3]. However, it is not realistic for users to manually specify their preferences for each possible use case. Furthermore, several studies have shown that users often set contradictory policies because they are influenced by contingent factors or by misunderstanding [4]. It is therefore necessary to identify artificial intelligence (AI) methods to support the user in understanding the privacy policies of services and app, and in suggesting the most appropriate choice regarding the sharing of private content.

In this paper, we address this challenge by proposing AI-based techniques for assisting the user in taking informed privacy decisions about new app/devices and data sharing in social networks. We propose a technique relying on machine learning and natural language processing to automatically parse privacy policies written in natural language, recognize the privacy-relevant entities in the text, and answer the user's questions about the privacy practices of the provider regarding certain private data types. We also present advanced feature extraction methods and a recognition algorithm to analyze user's images in order to suggest the most appropriate sharing option. We experimented with our techniques on real-world datasets, and the results show the effectiveness of our methods. We also developed a preliminary prototype of an envisioned system for the integrated support of users' privacy choices, which includes a chatbot that can communicate in natural language with the user.

The main contributions of our work are the following.

- We present machine learning techniques for retrieving and classifying paragraphs related to the privacy practices of 30 data types from natural language privacy policies;
- We propose artificial intelligence methods to automatically assess the level of privacy risk determined by sharing personal images in social networks;
- We present an experimental evaluation of our system with real-world data;
- We present ongoing work on a preliminary prototype of a system for supporting user privacy choices with regard to online social networks.

The rest of the paper is structured as follows. Section 2 illustrates the related work about automatic tools for information retrieval from privacy policies, and privacy methods for image sharing in social networks. Section 3 explains the technique for retrieving natural language information from raw privacy policies. Section 4 presents the methods used to extract data and meta-data from images, and the techniques to determine their level of privacy risk. Section 5 shows our experimental results. Section 6 presents ongoing work on a system prototype and chatbot for supporting privacy choices. Section 7 concludes the paper and outlines promising directions for future work.

2. Related Work

In this section, we present related work on information retrieval from privacy policies, and on privacy methods for image sharing in social networks.

2.1. Automatic Tools for Information Retrieval from Privacy Policies

As anticipated, in this work, we aimed to devise human-friendly techniques for supporting the user in setting their privacy policies regarding apps and smart devices. Hence, our work relies on underlying techniques to automatically parse and extract privacy concepts from privacy policies, which are normally written in natural language.

Brodie et al. proposed SPARCLE, an online workbench for generating machine-readable privacy policies [5]. Unlike traditional privacy policies written in natural language only, SPARCLE can transform the policies into XML format, enabling automatic processing. XML generation is based on different strategies applying natural language analysis methods and using a set of grammars to identify the dominant elements in privacy policy practices. Those methods may be used not only to generate XML representations, but also to automatically retrieve structured information from natural language privacy policies.

Bathia and Breaux proposed a method to develop a lexicon of data types for privacy policies [6]. Initially, they created a list of data types extracted from the paragraphs of 15 privacy policies by crowd workers, who provided 3850 annotations. Based on the analysis of the part-of-speech (POS) of those annotations, they used natural language processing techniques to identify reliable POS patterns to be reused for extracting privacy-relevant entities from natural language privacy policies. They identified 725 unique entities (e.g., 'email address', 'gender', etc.) that can be used to automatically retrieve information from the textual description of privacy policies. In our work, we use a similar method to obtain a vocabulary of privacy-relevant key terms, as explained in Section 3.1.2.

Zimmeck and Bellovin addressed the classification of privacy policy texts using ML techniques, in order to enhance transparency in online browsing [7]. As part of that work, they proposed a system called Privee as an extension of Google Chrome providing a summary of the privacy practices of the visited website. Narouei et al. proposed a method to extract the access control policies from natural language documents using semantic role labeling [8], while Spruit et al. used natural language processing techniques to extract policies from internal bank documents [9].

Zimmeck et al. [10] addressed the problem of analyzing and verifying the correspondence between the practices described in the privacy policy and the data actually collected and processed by mobile apps. They created an 'app privacy policy corpus' (named APP-350) annotated by legal experts and freely available on the Web. APP-350 was used to create machine learning (ML) models in order to understand which paragraphs of the privacy policy contain *privacy practices*: i.e., those parts of the privacy policy that describe which kind of personal data is collected/used, by whom (first or third parties), and how it is used. The trained ML models were used to analyze the texts of over 1,000,000 apps available on the Google Play Store, and to verify their effective implementation: e.g., checking the required permits, the included libraries, and the use of the Android APIs, finding conspicuous discrepancies in a relevant portion of apps.

Fan et al. proposed an automatic system for mining the privacy policies of apps and check the compliance with the European General Data Protection Regulation (GDPR) [11]. Based on a dataset of about 800 apps, they found that a relevant portion of privacy policies are incomplete, contain inconsistencies, or reveal serious issues, such as the use of insecure communication channels.

Those results motivate the need for automatic tools for assessing privacy policies in mobile computing and for supporting the user in taking informed decisions about privacy policies.

2.2. Privacy Methods for Image Sharing in Social Networks

Zerr et al. collected a dataset of images labeled according to their privacy level [12]. They designed a game where the participants were given a set of images and had to decide whether each image should be marked as 'private' or 'public'. Altogether, they collected more than 37,000 images publicly available from Flickr (https://www.flickr.com/, accessed on 20 May 2021) which had at least five tags each describing the image content. We used that dataset to experimentally evaluate our methods for automatically determining the privacy level of images shared on social networks.

Different methods were devised to automatically determine the privacy level of an image. Usually, they use standard machine learning algorithms, and perform classification based on different types of features extracted from the image. We can broadly classify feature types into visual features or textual features. The former extract characteristics from the visual content of the image. Face detection features rely on computer vision tools to detect the occurrence of faces in a picture. For instance, using the Viola–Jones face detector [13], it is possible to extract two features: i.e., the number of faces detected in the image, and a vector of the bounding boxes around each detected face. The presence of people's faces could be a hint of the sensitivity of a picture [14].

Zerr et al. hypothesized that the distribution of colors may be an indicator of the degree of sensitivity of an image [12]. They computed the histogram of an image based on HS (i.e., Hue, Saturation) color space: each of the two color dimensions was split into sixteen evenly divided segments, resulting in a histogram consisting of 32 bins. They observed that public images tend to have fewer but more intense colors, whereas private images tend to have more colors which are evenly distributed across the histogram.

Previous research showed that edge-based features could be used to discriminate outdoor vs. indoor images [15,16]. It was found that private images tend to contain strong, straight and almost vertical edges, associated with indoor environments, whereas public images had shorter and weaker edges [15]. Since the location where a picture was taken may determine its privacy level, this is an important feature to consider.

The presence of certain objects in an image may be useful to represent its context, which in turn can have an impact on the sensitivity of the picture. Object detection can be performed using different methods. The scale invariant feature transform (SIFT) descriptor uses a 128-dimensional vector to describe a texture found in an image. Each feature obtained through SIFT is then put inside a vocabulary of visual terms. Interesting image regions can be recognized through the detection of peaks in a difference of a Gaussian pyramid [17]. Different authors, including Squicciarini et al. [14] and Yang et al. [18], have used these features for inferring the sensitivity level of images. To have a better insight into the presence of faces, in this work, we consider not only the number of recognized faces, but also the ratio of the image that they cover, in order to assess whether they are prevalent or not.

Images shared on social networks are often tagged with user-defined keywords and comments. Previous works tried to extract textual content from tags in order to recognize the image privacy level, using methods normally used for text classification. It was found that, to some extent, there is correlation between the occurrence of certain tags and the sensitivity of the associated image [12]. Tonge and Caragea used deep visual semantic features derived from layers of convolutional neural networks, as well as textual features, to infer the privacy level of images [19]. Deep learning methods are also used by Yu et al. to suggest privacy settings for social image sharing considering both images' sensitiveness and trustworthiness of the viewers [20]. In our work, we experimented with different kinds of features, both in isolation and in combination, to improve the recognition rate of image privacy level detection.

3. Information Retrieval for Privacy Policies

In this section, we illustrate the technique for retrieving natural language (NL) information from raw privacy policies.

Whenever they provide an online service, IT companies must publish a legal document written in natural language to describe what kind of personal data they will collect, and how they will use such data. Normally, this document, called *privacy policy*, is composed of many paragraphs, which could contain *practices descriptions* such as the collection and use of data types (e.g., 'We collect identifier information from your device'), pledge of not collecting certain data types (e.g., 'We do not store any contact information with third parties'), or both. Certain paragraphs could also not contain any practice description at all. These privacy policies are then published in organization's websites to make them available to customers. In the following, we define key concepts that will be used in the rest of the paper.

- A *data type* is an attribute of personal information appearing in a privacy policy (e.g., 'location information' is the data type that describes the user's geographical location);
- A *privacy practice* is an activity of accessing, collecting, or sharing personal information of a given data type;

- A *privacy practice description* is a statement written in natural language in a privacy policy, which explicitly indicates whether certain privacy practices are being performed or not by the service provider.

Below we show an excerpt of a real-world privacy policy. The policy contains two privacy practice descriptions (highlighted in bold).

Example 1. *"In some circumstances, **Google also collects information about you from publicly accessible sources**. For example, if your name appears in your local newspaper, Google's Search engine may index that article and display it to other people if they search for your name. **We may also collect information about you from trusted partners, including marketing partners who provide us with information about potential customers of our business services, and security partners who provide us with information to protect against abuse**." [21]*

Of course, privacy violations can also occur as a consequence of sharing multimedia objects on social networks. Most online social networks allow users to define the visibility of shared contents. In particular, *private* contents are visible only to users that are explicitly allowed by the user (e.g., because the are in a friendship relationship), while *public* contents are visible to everyone. Incorrect settings in terms of the visibility of multimedia content may easily determine serious privacy issues [22].

For the sake of this work, we consider each paragraph in a privacy policy as a unit of information to be classified. Each paragraph may either contain a practice description or not. Each practice description may be related to one or more data types. Hence, we treat this problem as a set of binary classification tasks. For each data type, we use a binary classifier to determine whether the paragraph contains a privacy practice which refers to that data type.

3.1. Feature Engineering

Of course, the features that we use to classify paragraphs are built based on the textual content of the paragraph. The features of the paragraphs are extracted using the bag of words (BoW) model [23]. In the bag of words model, each word corresponds to a feature, and the feature value is computed based on the number of occurrences of the respective word. To build this structure, we use the *term frequency–inverse document frequency* (TF–IDF) [23] vector representation of the document, as defined below. Note that in our case, the document corresponds to a paragraph.

- Term frequency formula:

$$tf(w) = \frac{doc.count(w)}{doc.total()}, \tag{1}$$

  where *doc.count(w)* is the number of occurrences of the word w in the document, and *doc.total()* is the total number of words in the document.
- TF–IDF formula:

$$idf(w) = \log \frac{|\mathrm{D}|}{|\{d : w \in D\}|}, \tag{2}$$

$$tfidf(w) = tf(w) \cdot idf(w), \tag{3}$$

  where $|D|$ is the total number of documents, and $|\{d : w \in D\}|$ is the number of documents containing w.

The TF–IDF value increases proportionally to the number of times the term is contained in the document (*tf* part), but grows in inverse proportion to the frequency of the term in the whole collection (*idf* part). Hence, very common words (such as 'the') are not necessarily considered more important (i.e., having a larger *tfidf* score) than less common ones such as 'email' or 'pii' (the latter stands for 'personally identifiable information').

However, many words that appear in a paragraph are irrelevant for the purpose of classification. Hence, it is necessary to only retain the relevant ones in order to avoid problems such as overfitting and noisy text. To this aim, we investigated two solutions, explained in Sections 3.1.1 and 3.1.2.

3.1.1. Using Bi-Grams and Feature Selection

In order to consider more significant terms, for building feature vectors, we consider not only single terms but also *bi-grams*. Bi-grams are couples of adjacent words appearing in a text. They can help obtain a more effective classification model, because some entities or concepts are naturally represented by two, such as 'personal information', 'email address', or 'collect data' [10]. Obviously, considering the single terms (i.e., *uni-grams*)in the previous examples would result in a complete loss of the concept semantics.

However, using bi-grams together with uni-grams increases the number of features extracted from text. Most importantly, not all of the bi-grams are useful for text classification.

Example 2. *Consider the paragraph: 'In some circumstances, Google also collects information about you from publicly accessible sources'. The uni-grams and bi-grams in the paragraph are: 'In', 'some', 'circumstances', 'Google', 'also', 'collects', 'information', 'about', 'you', 'from', 'publicly', 'accessible', 'sources' and 'In some', 'some circumstances', 'circumstances Google', 'Google also', 'also collects', 'collects information', 'information about', 'about you', 'you from', 'from publicly', 'publicly accessible', 'accessible sources'.*

In the above example, bi-grams like 'you from' and 'circumstances Google' are not significant and should be discarded because they do not represent any entity or concept. Indeed, they are meaningless when read out of the paragraph context. Moreover, the construction of a feature vector with uni-grams and bi-grams extracted from an entire set of privacy policies would generate a very large number of features. This fact may generate different problems, described below.

- Overfitting: the classification model only learns to correctly classify training instances (labeled samples) but fails to classify new unknown instances;
- The text is noisy: frequently, texts obtained from the Web contain special characters, links, dates, numbers, or terms that do not add meaningful information to the classification task;
- Stopwords: many words in the document do not add useful information for classification, and unnecessarily increase the size of the feature vectors. These words are called 'stopwords'. Examples of stopwords are: 'the', 'at', 'that'.

To remove unneeded features and reduce the set of features, we applied the following text preprocessing steps.

- Using regular expressions, we removed numbers, urls, and dates from the original text;
- We removed stopwords using the nltk standard stopword list [24];
- We applied *stemming*: i.e., the process of reducing words to their root. For instance, *collected* is transformed to *collect*, *computation* is transformed to *comput*.

3.1.2. Using Vocabulary of Key Terms for Privacy Practices

As an alternative approach, we also used a hand-crafted dictionary of key terms as features, removing from the original text those words that do not belong to the dictionary, and applying stemming. Since different data types may be characterized by different words, we used specific key term dictionaries for each data type.

Of course, the classification results depend on the completeness and accuracy of the dictionary. Since we are not aware of any publicly available dictionary of key terms for privacy policies, we built the dictionaries from scratch using a corpus of privacy policies, manually choosing those terms that seem more related to the data type.

3.2. *Classification of Privacy Practices*

As explained before, each paragraph may contain no privacy practice, or a privacy practice referring to one or more data types. As a consequence, the problem we are tackling is a multi-label classification problem, in which the label/classes correspond to the data types.

In the literature, multi-label problems have been addressed in different ways [25]. A straightforward approach is to transform the multi-label problem into a single-label multi-class classification problem, where the classes are all the possible label combinations. However, for the sake of this work, we consider the 30 classes (i.e., data types); hence, with this approach, the number of possible classes would be 2^{30} = 1,073,741,824. Since the number of classes would be unfeasible, we discarded this approach.

In our work, we used the so-called *binary relevance method*, which is widely used in the literature [26]. Hence, we decomposed the multi-label problem into multiple binary classification tasks, one for each data type. Every data type classifier uses two classes: ‘yes’, meaning the presence of a privacy practice description regarding that data type; and ‘no’ meaning the opposite. In Table 1, we list the considered data types, each corresponding to a binary classifier. For the sake of this paper, we used a random forest classifier, since it is generally effective for text classification [27].

Table 1. Privacy practice data types.

Data Type	Description	Example
Contact	User’s generic contact data privacy practices	We collect some contact information from you
Contact address book	Regarding the user’s phone address book	We could access to your contact list such as phone address book
Contact city	User’s city	We collect some contact information such as your city
Contact email address	User’s email address	When you subscribe, you provide us some of your personal information such as your email
Contact password	User’s password	When you subscribe, you provide us some of your personal information such as your account password
Contact phone number	User’s phone number	When you subscribe to our service, you provide us some of your personal information such as your phone number
Contact postal address	User’s postal address	We collect some contact information such as your postal address
Contact ZIP	User’s ZIP (Zoning Improvement Plan) code	We collect some contact information such as your ZIP code
Demographic	User’s unspecified demographic data	We collect some demographic information from you
Demographic age	User’s age	We collect some demographic information from you such as your age
Demographic gender	User’s gender	We collect some demographic information from you such as your gender
Identifier	Unspecified identifiers	We may collect usage information; this can include some device identifiers’ information
Identifier ad ID	Identifier for advertising	We could collect some advertising identifier information such as your Google ad ID (gaid)
Identifier cookie or similar tech	Cookies, pixel tags, etc.	Cookies are unique identifiers that we transfer to your device to enable our systems to recognize your device
Identifier device ID	Device identifiers	
Identifier IMEI	IMEI code	We could collect some device identifier information such as your device IMEI code

Table 1. *Cont.*

Data Type	Description	Example
Identifier IMSI	IMSI code	We could collect some device identifier information such as your device IMSI code
Identifier IP address	IP address as identifier	Information like technical properties and general usage information such as IP address may be processed
Identifier MAC	Device's MAC address	We could collect device ID information such as your device MAC Address
Identifier mobile carrier	Mobile carrier	
Identifier SIM serial	User's SIM number	We could collect SIM identifier information such as your SIM serial number
Identifier SSID BSSID	SSID and BSSID	We could collect network ID information such as your WLAN SSID/BSSID code
Location	Unspecified data about device location	We collect some information about your location
Location Bluetooth	Device location through bluetooth	We collect some information about your location based on Bluetooth
Location cell tower	Device location through cell towers	We collect some information about your location based on cell towers
Location GPS	Device location through GPS	We collect some information about your location based on GPS
Location IP address	Device location through IP address	We collect some information about your location based on IP address
Location WiFi	Device location through WiFi	We collect some information about your location based on WiFi
SSO	Single sign on information from unspecified service	You allow us to collect (or the third party to share) information about you
Facebook SSO	Single sign on information from Facebook service	You allow the third party (e.g., Facebook) to share information about you

4. Image Privacy Recommendations

In this chapter, we describe the methods used to extract data and meta-data from images, and the techniques to determine their privacy level.

4.1. Numerical Feature Extraction

We adopted different methods to extract numerical data and meta-data from images, with the goal of building feature vectors for machine learning classification.

4.1.1. Hue–Saturation Data

A digital image can be, at a lower level, represented by a matrix of points: each point holds one or more values that can be represented as a tuple. Those tuples belong to a given **color space**, such as *RGB* (red–green–blue), *CMYK* (cyan–magenta–yellow–black), etc. For extracting these features, we use the *HSV* (hue–saturation value) color space, as it separates the color from its intensity (the value). For each image, we extract the hue and saturation histograms, which are divided into 16 bins each. The values inside the bins are normalized using the number of pixels of the image to prevent values growing proportionally to the image size.

4.1.2. Face Detection

Face detection algorithms are computer vision tools that allow recognizing and locating faces within pictures. Since the presence of people may easily determine the sensibility level of an image, these are important features to consider. In our work, face detection is implemented using the Python library written by Adam Geitgey (https:

//www.github.com/ageitgey/face_recognition, accessed on 20 May 2021). In order to retain information about the number and size of recognized faces, we also considered the ratio between the area covered by the faces and the total area of the image.

4.1.3. Scale Invariant Feature Transform and Bag of Visual Words

In previous works, scale invariant feature transform (SIFT) features proved to be useful when used to find different points of interest inside an image [12,14,28], as well as effective features to predict the privacy level. SIFT methods identify characteristic visual features that are useful for different computer vision tasks, including classification and object recognition.

To reduce the size of the data points, as done by Montazer et al. in [29], we compressed the SIFT feature vector to an eight-dimensional feature vector. In previous works, this compression method retained the quality of the features for classification [29]. Then, we used a variation of the *bag of words* (BoW) model, which we named *bag of visual words*.

This essentially consists of a series of visual terms which could be found inside the image. The occurrence of one or more visual terms is used as an image feature. Each visual term is composed of clustering different (compressed) visual features, so that the newly found cluster is the visual term itself. Finding the occurrence of a visual term for a given image is simply a matter of finding which cluster a visual term belongs to. We used a visual word vocabulary of 1000 visual terms.

4.2. Textual Feature Extraction

In the following, we explain the methods used for extracting textual data associated to the images for classification.

4.2.1. Weighted Graph

Weighted graphs were used in [30] for classifying the images using associated tags (keywords) that describe their content. Since images could have tags in common, a graph could be built to highlight the relationships between words and the associated privacy policies. Previous research showed that the image tags tend to be noisy [31]. Hence, two different but semantically similar words which could provide meaningful insight into building a context tend to be separated in the graph. In order to avoid this problem, we apply stemming to the tags.

After stemming the words, we built the following graph. Vertices are represented by the stemmed tags, while edges represent the occurrence of two tags in the tag list for a given image. Edges are non-oriented and weighted. For each occurrence of two tags in the same list for a given image, the weight of the edge connecting them is incremented by 1. The weight is useful to aggregate the vertices and find groups of semantically close tags.

A group of words might provide insight into the decision of a privacy policy. Hence, we aggregated the vertices in different clusters using the Louvain method [30,32]. Each tag has a privacy policy associated to the image where it was first found in. For a given group of vertices, the most frequent privacy policy was chosen as the representative.

For each tag, we used a relevance measure, similarly to what was proposed by Squicciarini et al. in [30]:

$$r_G(t) = \begin{cases} 0 & \text{if } |C_t| \leq 1, \\ \frac{deg_{C_t}(v_t)}{|C_t|\cdot(|C_t|-1)\cdot\frac{1}{2}} & \text{otherwise.} \end{cases} \tag{4}$$

Here, $deg_{C_t}(v_t)$ represents the number of edges in the cluster C_t incident to the vertex $v_t \in C_t$ associated to a tag t, and $|C_t|$ is the number of vertices in the cluster for a given v associated to t.

The above equation provides a measure for the relevance of a word in evaluating the appropriate privacy policy. It represents the ratio of existing connections inside a group for a given tag (vertex) with respect to the maximum number of links (edges) inside the group. The equation can be interpreted as follows:

- If the stemmed tag does not belong to any group, it cannot be evaluated;
- If the group is made up by the stemmed tag itself, it might have no meaning and it should not be considered;
- If the stemmed tag (vertex) is linked to at least one other stemmed tag inside the group, the relevance measure can be calculated.

Based on the equation, it is easy to consider all the tags representing an image and their associated (weighted) privacy policy values in order to compute the privacy policy (i.e., public or private) of the image based on a weighted sum.

4.2.2. Bag of Words Model

With this method, we built a dictionary of terms, namely the tags, and for each image we checked whether a tag occurred in its tag list. At first, we built the tags dictionary by simply putting the different terms appearing in the whole picture dataset in a set. Each tag corresponds to a feature. Then, we used the standard BoW method (explained in Section 3.1) to create a feature vector for each image. The resulting dataset was used with the chosen classifier to determine the image privacy level.

5. Experimental Evaluation

In this section, we illustrate our experimental results. The objective is to evaluate the effectiveness of our analysis methods for policies and images with real-world data. We performed the experiments for information retrieval from privacy policies, presented in Section 5.1, using a large dataset of 350 privacy policies. Each policy was manually labeled at the sentence-level with the kind of involved data. The objective of those experiments was to assess the accuracy of our algorithm for privacy policy knowledge extraction. For the experiments about the privacy risk analysis of images, presented in Section 5.2, we used a dataset of 28,000 images extracted from a well-known image sharing social network. Each image was manually tagged with the involved privacy risk level. The objective of those experiments was to assess the effectiveness of our algorithm in recognizing the correct privacy level risk of images.

5.1. Information Retrieval from Privacy Policies

In the following, we showed the results of the two solutions explored in this work. We recall that the classification task was to recognize, for each data type, whether the paragraph of a raw NL privacy policy refers to it (classed true) or not (classed false). Hence, for each classification task, four cases can occur:

- A paragraph that should be classified as true is classified as true. These instances are called true positives (TPs);
- A paragraph that should be classified as true is classified as False. These instances are called false negatives (FNs);
- A paragraph that should be classified as false is classified as true. These instances are called false positives (FPs);
- A paragraph that should be classified as false is classified as false. These instances are called true negatives (TNs).

5.1.1. Privacy Policy Dataset

In order to experiment our techniques, we used a large annotated corpus of privacy policies. The dataset, called APP-350, was developed within the MAPS project [10]. The dataset consists of a set of 350 privacy policy text files. All files were subdivided into paragraphs, each one annotated by law students. Every annotation consists of a tag, applied to paragraphs, which indicates that the paragraph contains a privacy policy description. For example, the *Contact* tag was attached to those paragraphs referring to contact information practice. A paragraph can also be tagged with different annotations at same time: e.g., the same paragraph may describe both demographic and contact privacy practices. In that corpus, five generic tags (i.e., contact; location; demographic; SSO = single sign on; identifier) were used, plus more specific sub-tags (e.g., ContactEmailAddress; ContactPhoneNumber…). In total, there were 30 different tags that could be applied to the paragraphs, which are listed in Table 1.

5.1.2. Experimental Setup

We used the python language to implement the techniques described in Section 3. The APP-350 corpus contains files in yaml format. The Python *PyYAML* library was used to parse and read the files contained in the corpus. The *nltk* library was used to stem the words contained in text and remove stopwords. The *re* (regular expression) standard Python library was used to remove dates, numbers, and urls from privacy policies texts. *Pandas* is a Python library that allows to easily handle a set of data represented in tables. We used Pandas to construct a data structure based on the information contained in the annotated dataset. We built a Pandas table (DataFragment) to store in rows of segments and columns all the annotations present in the APP-350 corpus. This structure indicates which annotations the paragraphs are tagged with. The TfIdfVectorizer from *scikit learn* library was used to construct the TF–IDF vector on the set of paragraphs. A row in the TF–IDF vector represents a paragraph and the columns are the TF–IDF features.

For performing machine learning, we used the Weka [33] library for Python, which comes with the implementations of the main classification algorithms. We chose to use random forest with default configuration, since it is known to be effective for text classification. We also chose to wrap the Random Forest with the Weka AttributeSelectedClassifier, which performs attribute selection on the features of the dataset to retain only those features that are useful to increase the classification results. The *liac-arff* library was used to save the training dataset in an sparse '.arff' file, which is the native file format of Weka.

We used 10-fold cross validation for evaluating our techniques on the APP-350 dataset. We considered the following metrics:

- Precision: ratio between the number of instances correctly classified as positive (TP) and the total number of instances classified as positive (TP + FP);
- Recall: ratio between the number of instances correctly classified as positive (TP) and the number of instances that are actually positive (TP + FN);
- F_1: harmonic mean of precision and recall;
- Accuracy: the ratio of correctly classified instances: i.e, $\frac{TP+TN}{TP+TN+FP+FN}$.

For each technique, we report the individual results for each specific sub-tag, and the aggregated results for generic tags (i.e., data types).

5.1.3. Results Using Bi-Grams and Feature Selection

Table 2 reports the classification result of each classification task. As we can see, the technique achieves good results, with an overall accuracy of 0.979 and F_1 score of 0.89. Generally, the recall values are lower than precision values: this means that the classifiers tend to miss the identification of few positive instances (i.e., relatively high number of FNs), but those identified as true instances are very often correct (i.e., low number of FPs). This result may be due to the fact that classes are strongly imbalanced. Indeed, the number of negative samples significantly overcomes the one of positive samples in all binary classification problems. Based on the F_1 score, the best results are reached by a few specific *Identifier* sub-tags, while the worst results are obtained by the generic identifier data type and its remaining sub-tags. It should be noted that the classification problems that we are tackling are rather challenging, because classes are strongly unbalanced. Indeed, the negative class (false) is much larger than the positive one (i.e., 92% vs. 8%, respectively).

Table 2. Classification performance using the bi-gram method.

Data Type	Precision	Recall	F_1	Accuracy
Contact	0.891	0.871	0.881	0.975
Contact address book	0.925	0.867	0.894	0.979
Contact city	0.973	0.894	0.93	0.996
Contact email address	0.932	0.924	0.928	0.941
Contact password	0.922	0.936	0.929	0.985
Contact phone number	0.959	0.901	0.927	0.965
Contact postal address	0.916	0.814	0.856	0.954
Contact ZIP	0.971	0.857	0.905	0.992
Demographic	0.931	0.92	0.925	0.987
Demographic age	0.959	0.917	0.937	0.984
Demographic gender	0.958	0.919	0.937	0.988
Identifier	0.857	0.626	0.683	0.973
Identifier ad ID	0.984	0.87	0.918	0.985
Identifier cookie or similar tech	0.967	0.958	0.962	0.97
Identifier device ID	0.931	0.885	0.906	0.958
Identifier IMEI	0.999	0.954	0.975	0.998
Identifier IMSI	1	0.893	0.94	0.999
Identifier IP address	0.978	0.943	0.96	0.977
Identifier MAC	0.976	0.917	0.944	0.993
Identifier mobile carrier	0.885	0.844	0.863	0.986
Identifier SIM serial	0.874	0.799	0.832	0.996
Identifier SSID BSSID	0.998	0.639	0.717	0.997
Location	0.942	0.923	0.932	0.953
Location Bluetooth	0.916	0.842	0.875	0.984
Location cell tower	0.936	0.795	0.851	0.983
Location GPS	0.956	0.867	0.906	0.981
Location IP address	0.93	0.731	0.799	0.977
Location WiFi	0.923	0.782	0.837	0.975
SSO	0.883	0.806	0.839	0.964
Facebook SSO	0.887	0.931	0.908	0.982
Aggregate contact	0.922	0.917	0.919	0.921
Aggregate location	0.953	0.927	0.94	0.957
Aggregate demographic	0.951	0.911	0.93	0.972
Aggregate identifier	0.944	0.942	0.942	0.942
Aggregate SSO	0.891	0.872	0.881	0.965
Aggregate all	0.939	0.861	0.89	0.979

5.1.4. Using Vocabulary of Key Terms for Privacy Practices

Table 3 shows the results achieved using the vocabulary of key terms for privacy practices. Overall, with respect to the use of bi-grams, this method achieves higher accuracy (0.991 vs. 0.979) and lower F_1 score (0.683 vs. 0.89). As we can also see with this method, precision is larger than recall. Based on the F_1 score, the best results are achieved for some identifier sub-tags, while the lowest results are obtained by *aggregate SSO*, *SSO*, and *contact* data types.

The accuracy achieved by this method is impressive. However, since the classes are strongly unbalanced, accuracy is not a particularly reliable metric to evaluate these results. Indeed, from the results, we can observe that the classifiers using the vocabulary of key terms are biased towards the most common class ’false’. As a consequence, they achieve high accuracy, but their results in terms of F_1 score are much lower. In contrast, the results achieved using bi-grams are less biased and achieve a good F_1 score.

Table 3. Classification performance using the vocabulary of key terms for privacy practices.

Data Type	Precision	Recall	F_1	Accuracy
Contact	0.495	0.213	0.298	0.985
Contact address book	0.736	0.299	0.425	0.988
Contact city	0.768	0.606	0.677	0.997
Contact email address	0.837	0.794	0.815	0.972
Contact password	0.797	0.671	0.729	0.993
Contact phone number	0.856	0.772	0.812	0.986
Contact postal address	0.741	0.476	0.579	0.982
Contact ZIP	0.9	0.686	0.778	0.997
Demographic	0.805	0.842	0.823	0.995
Demographic age	0.914	0.741	0.819	0.994
Demographic gender	0.922	0.809	0.862	0.996
Identifier	0.701	0.353	0.47	0.993
Identifier AD ID	0.838	0.4	0.542	0.990
Identifier cookie	0.873	0.892	0.883	0.982
Identifier device ID	0.83	0.771	0.8	0.986
Identifier IMEI	0.946	0.897	0.921	0.999
Identifier IMSI	1	0.786	0.88	1.000
Identifier IP address	0.952	0.829	0.886	0.990
Identifier MAC	0.958	0.827	0.888	0.998
Identifier mobile carrier	0.833	0.495	0.621	0.996
Identifier SIM serial	0.762	0.533	0.627	0.999
Identifier SSID BSSID	0.636	0.389	0.483	0.999
Location	0.833	0.796	0.814	0.978
Location Bluetooth	0.795	0.682	0.735	0.995
Location cell tower	0.792	0.421	0.55	0.994
Location GPS	0.871	0.706	0.78	0.994
Location IP address	0.719	0.403	0.516	0.992
Location WiFi	0.779	0.595	0.674	0.993
SSO	0.508	0.23	0.317	0.982
Facebook SSO	0.606	0.417	0.494	0.989
Aggregate contact	0.766	0.565	0.639	0.988
Aggregate location	0.798	0.601	0.678	0.991
Aggregate demographic	0.88	0.797	0.835	0.995
Aggregate identifier	0.863	0.682	0.753	0.994
Aggregate SSO	0.557	0.324	0.406	0.986
Aggregate all	0.8	0.611	0.683	0.991

5.2. Prediction of Image Privacy Level

In the following, we report the experimental results about the prediction of image privacy level.

5.2.1. Image Privacy Level Dataset

In order to evaluate our methods, explained in Section 4, we used the *PicAlert* (http://l3s.de/picalert/, accessed on 20 May 2021) dataset made available by Zerr et al. [12]. The dataset contains Flickr images and metadata, including the privacy level of each image, that can be: *public*, *private*, or *undecidable*. The privacy level was manually assigned by individuals inspecting each image and evaluating how much its release may harm the privacy of the owner. More details about the labeling procedure and dataset may be found in the original paper [12]. For the sake of our study, we considered two privacy levels: public and private. The dataset contains only the url of pictures, not the pictures themselves. Hence, we downloaded from Flickr all those pictures that were still available online at the time of writing. We collected a dataset of about 28,000 images, in which the distribution of public and private ones was homogeneous.

5.2.2. Experimental Setup

The PicAlert dataset also contains *keywords* assigned by users, which we used as tags for building the BoW model described in Section 4.2.2. We also applied the techniques described in Section 4 to extract the other features of the images. We applied 10-fold cross validation in all the experiments. Results were evaluated in terms of precision, recall, F_1 score, and MCC. The latter is the Matthew's correlation coefficient, a measure which assumes values between -1 and 1, where 1 indicates perfect correlation between the predictions and the ground truth, 0 indicates no correlation, and -1 indicates perfect negative correlation. We performed experiments using support vector machines (SVMs) and random forest classifiers, since they are among the most effective ones for this task.

5.2.3. Results

At first, we evaluated the effectiveness of privacy level prediction considering one kind of feature in isolation, and then considered them in conjunction.

Hue–Saturation Features

In a first experiment, we evaluated the feature extraction method described in Section 4.1.1. Results are shown in Table 4.

Table 4. Results of image privacy level prediction using hue–saturation features.

Classifier	Precision	Recall	F_1	MCC
SVM	0.58	0.581	0.581	0.162
Random forest	0.614	0.613	0.613	0.227

The best results were obtained using random forest, considering the values of precision, recall, F_1 score, and MCC. However, the results were only slightly better than the ones achieved by a random classifier, meaning that hue–saturation features alone are not sufficient for reliably recognizing the privacy level of images.

Face Detection Features

In a second experiment, we evaluated the use of face detection features, described in Section 4.1.2. The achieved results are shown in Table 5.

Table 5. Results of image privacy level prediction using face detection features.

Classifier	Precision	Recall	F_1	MCC
SVM	0.674	0.655	0.645	0.328
Random forest	0.726	0.686	0.670	0.409

Previous research showed that the presence of faces inside images are a good hint to determine the privacy level. As expected, results with these features improved with respect to the use of hue–saturation features. In this case, the random forest classifier achieved the best results, obtaining an F_1 score of 0.67. Moreover, the MCC value of around 0.40 clearly indicates a significant correlation between the predictions and the ground truth. Nonetheless, by inspecting the image dataset, we noticed that the presence of faces was not always sufficient to distinguish private vs. public images, since (even without faces or human presence) other elements in the picture may determine privacy issues. Hence, face detection features are not sufficient by themselves for this classification task.

Bag of Visual Words

The SVM classifier has proven not to be efficient for this case. The bag of visual words, as introduced in Section 4.2.2, can be considered a dictionary of visual terms derived from clustering SIFT features. Results with these features are shown in Table 6.

Table 6. Results of image privacy level prediction using bag of visual words features.

Classifier	Precision	Recall	F_1	MCC
SVM	0.589	0.584	0.577	0.173
Random Forest	0.615	0.614	0.613	0.229

Random forest also obtained the best results in this case. However, as with hue–saturation features, the results are only marginally better than those that would be obtained by a random classifier.

Bag of Words

Finally, we evaluated the use of the bag of words method explained in Section 4.2.2. Results are shown in Table 7.

Table 7. Results of image privacy level prediction using bag of words features.

Classifier	Precision	Recall	F_1	MCC
Random Forest	0.600	0.593	0.586	0.192
SVM	0.573	0.573	0.573	0.167

The random forest classifier also achieved the best results when using these features. However, the results are not positive. Hence, when used in isolation, BoW features are not effective in determining the privacy level of images.

Considering More Features at Once

As shown in previous results, all features have a correlation with the ground truth. Hence, we tried different a combination of features to improve the recognition rates. In these experiments, we used the Random Forest classifier, since it achieved the best results in all the previous experiments.

Results are shown in Table 8. We report those combinations obtaining large recognition rates. The best results were achieved using both hue–saturation features and face detection ones. Results improved significantly with respect to the use of the two kinds of features separately. Indeed, the achieved F_1 score is 0.727 (vs. 0.67 achieved by face detection features alone), and the MCC value is 0.473 (vs. 0.409 achieved by face detection features alone). In particular, with the combined features, we noticed a remarkable increase in recall: i.e., recall is 0.731 with the combined features, while it is 0.686 using face detection features alone. These results show that the combination of different feature types, both considering the presence of people faces and low-level characteristics of the image, are needed to improve the recognition rates. Overall, we observed that the combination of different features improves the recognition rates with respect to the use of single types of features.

Table 8. The results of image privacy level prediction using different feature combinations and the random forest classifier.

Features	Precision	Recall	F_1	MCC
HS + face detection	0.743	0.731	0.727	0.473
HS + SIFT	0.621	0.621	0.621	0.242
Face detection + BoW	0.712	0.709	0.707	0.421
Face detection + SIFT	0.654	0.654	0.654	0.308
Face detection + BoW + SIFT	0.635	0.633	0.632	0.268
Face detection + BoW + HS	0.725	0.723	0.723	0.448

Weighted Graph

Finally, we experimented the weighted graph method explained in Section 4.2.1. In this experiment, we used only those images having some associated tags, resulting in a dataset of 9300 images well balanced between public and private ones. Results are shown in Table 9.

Table 9. Results of image privacy level prediction using the weighted graph method.

Private Images F_1	**Public Images F_1**	**F_1**	**Accuracy**
0.628	0.620	0.624	0.624

In general, the results achieved by the weighted graph method were inferior to the ones obtained using numerical features. Of course, the effectiveness of this method depends on the accuracy of tags. With more accurate tags and a larger knowledge base, we expect achieving better results.

6. Ongoing Work

In this section, we present ongoing work on a preliminary prototype of an integrated system to assist the user in taking privacy choices for the release of personal contents online.

6.1. Overview

Figure 1 shows an overview of the system prototype. The user communicates in natural language (NL) with a conversational agent, or *chatbot*. The chatbot can be executed on multiple personal devices, including workstations, smartphones, or virtual assistants. The goal of the chatbot is to provide the user with advice about the most appropriate decisions to take regarding their privacy policies and multimedia sharing practices.

Each time the user installs a new app, or plugs a new smart device, the chatbot contacts the AI-powered privacy agent for acquiring a NL description of the app/device privacy policies. The agent queries the NL policy description database to obtain the requested descriptions. Those descriptions are obtained applying natural language processing (NLP) methods to raw privacy policies retrieved from the Web using the Google Custom Search APIs. The agent makes use of machine learning (ML) algorithms to understand which paragraphs of the raw privacy policy contain practice descriptions.

The agent uses an annotated dataset of privacy policies to train the ML models. In particular, it trains a different classifier for each data type. A feature engineering process is used for generating a feature vector for each paragraph, removing unnecessary data such as stopwords. Once the structured information is extracted from paragraphs, the chatbot generates a synthetic description of the privacy policy of the app/device and communicates it in NL to the user. The chatbot may also answer a user's questions regarding specific aspects of the policy, such as the intended use of certain kinds of personal information, or the sharing policies about particular data.

Moreover, when the user wants to publish a new personal image on a social network, they may ask the chatbot about the suggested privacy level of the image. The chatbot queries the AI powered privacy agent, which forwards the image to the image privacy reasoner. The latter is in charge of extracting a set of features regarding the image, including face detection features, hue–saturation data, scale-invariant features, as well as tags describing the image content. Tags and face-detection features are extracted using computer vision techniques. The feature vector is provided to the ML image reasoner, which classifies the image as either public or private. The ML algorithm uses a model trained on a dataset of labeled images. The predicted label is communicated to the chatbot, which suggests the most appropriate privacy level to the user.

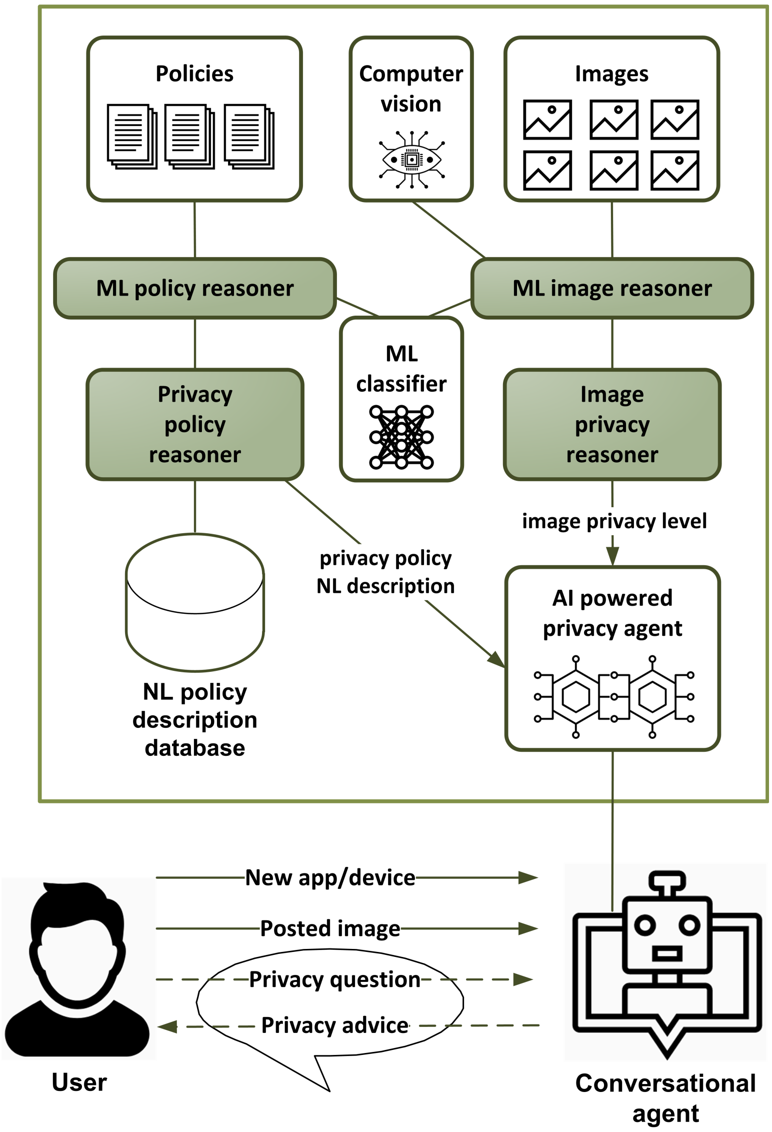


Figure 1. Prototype overview.

6.2. Chatbot Prototype

We developed a prototype of the conversational agent to assist the user in taking privacy decisions and in inspecting and querying privacy policies. An experimental evaluation of the prototype will be carried out in future work. We used the Google DialogFlow (https://dialogflow.cloud.google.com, accessed on 20 May 2021) the platform for developing the chatbot, and the *"Terms of Service; Didn't Read"* (ToS;DR) APIs to retrieve raw privacy policies. ToS;DR is a community which aims to analyze and evaluate the terms of service and privacy policies of sites and services. The site exposes an API for retrieving more information about a site or a service such as links to the privacy policies' pages. We used these links to obtain the privacy policy's texts with a Web scraper in order to obtain the privacy policies of the user's new app or devices.

In Dialogflow, the basic conversation is structured according to this pattern:

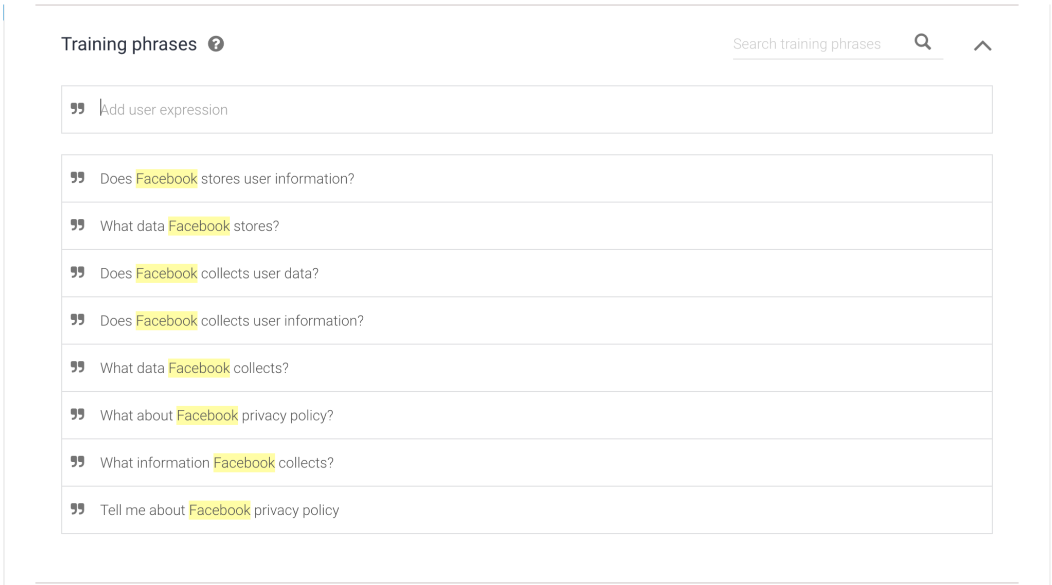
1. Input from user;
2. Input parsing by the DialogFlow agent;
3. Output response to the user.

To define the structure of conversation, DialogFlow provides intents that, based on user inputs, create responses. The main fields in the intents are:

- *Training phrases*, that can activate the intent upon user request, determine what to extract from these expressions, and how to respond. An intent represents a single stage or phase of a conversation;
- *Fulfillment*, which is code deployed as a webhook and allows to generate dynamic responses or trigger actions on a back-end;
- *Action* passed to a fulfillment—the purpose of this component is to activate a specific application logic;
- *Parameters*: in training phrases, the developer can highlight entities to be extracted as parameters. This allows to build structured data from user input. These data can be used to execute some application logic, or to generate semi-dynamic responses.

Figure 2 shows an example of intent and training phrases in our prototype. Thanks to the learning abilities of the framework, the user does not need to use a particular protocol or structured language to ask information to the chatbot, and can communicate with the agent in natural language. We structured the conversation with the agent in three steps:

1. The user requests information regarding a privacy policy;
2. The user requests information about a specific data type;
3. The user requests to inspect the paragraphs of the privacy policy text related to that specific data type.



Does Facebook stores user information?

What data Facebook stores?

Does Facebook collects user data?

Does Facebook collects user information?

What data Facebook collects?

What about Facebook privacy policy?

What information Facebook collects?

Tell me about Facebook privacy policy

Figure 2. Training phrases in an intent. Entities to be extracted are highlighted.

An intent has been created for each of these steps. In the first step, the user requests information about an app/device privacy policy. If that policy is already available in the classification result file, the agent creates a dynamic response with a list of generic data types privacy policies regarding the app/device. After this step, the user can:

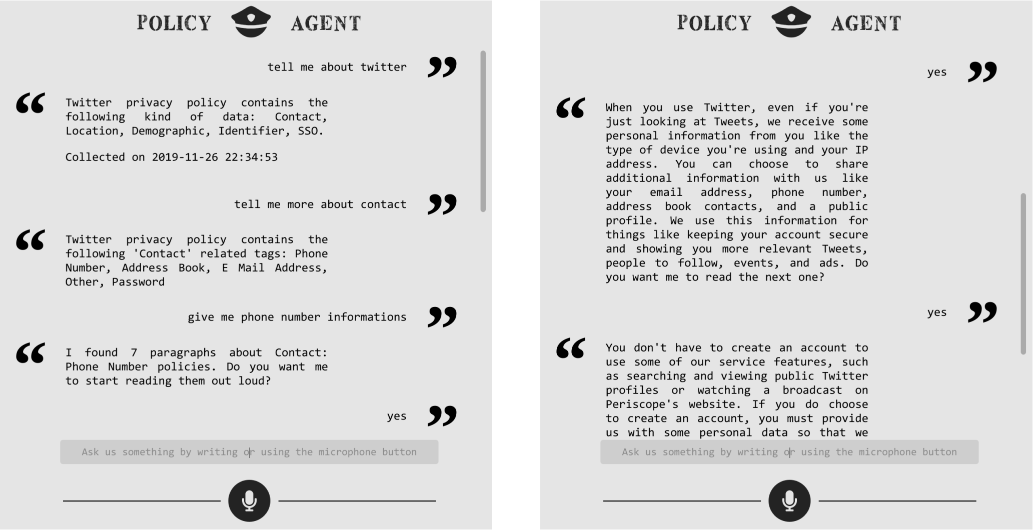
- Ask for another app/device privacy policy; or
- Ask for more information about a specific data type of the same app/device.

If the users requests more information about a specific data type, the agent will respond by enumerating all the sub categories of the requested data type. Finally, the user can request the original text that contains the privacy practice descriptions of the previously requested data type. The chatbot responds by reading the privacy practices of interest according to the classification of the raw privacy policy paragraphs, as explained in Section 3.

A different entity is used to retrieve an image to be published on social networks, analyze it according to the technique explained in Section 4, and suggest the appropriate privacy level to the user.

All of these steps make use of fulfillments. The agent is hosted by the DialogFlow cloud console. In order to produce dynamic responses, we built a Web app using *Flask* (https://flask.palletsprojects.com, accessed on 20 May 2021), that is a lightweight framework for Web development. Fulfillment in DialogFlow is implemented through webhooks and HTTPS requests.

Figure 3 shows an example of the session during which the user asks the chatbot about the privacy policies of Twitter, asking for further information about the way the service handles contacts, and in particular phone number information. Of course, the user can interact vocally with the agent using a smart speaker like Google Nest.



tell me about twitter

Twitter privacy policy contains the following kind of data: Contact, Location, Demographic, Identifier, SSO.

Collected on 2019-11-26 22:34:53

tell me more about contact

Twitter privacy policy contains the following 'Contact' related tags: Phone Number, Address Book, E Mail Address, Other, Password

give me phone number informations

I found 7 paragraphs about Contact: Phone Number policies. Do you want me to start reading them out loud?

yes

yes

When you use Twitter, even if you're just looking at Tweets, we receive some personal information from you like the type of device you're using and your IP address. You can choose to share additional information with us like your email address, phone number, address book contacts, and a public profile. We use this information for things like keeping your account secure and showing you more relevant Tweets, people to follow, events, and ads. Do you want me to read the next one?

yes

You don't have to create an account to use some of our service features, such as searching and viewing public Twitter profiles or watching a broadcast on Periscope's website. If you do choose to create an account, you must provide us with some personal data so that we

Figure 3. Chatbot prototype.

7. Conclusions and Future Work

In this paper, we introduced a novel platform for assisting the user in setting up their privacy policies and taking informed privacy decisions. We presented techniques for retrieving and classifying paragraphs related to the privacy practices of 30 different kinds from raw privacy policy text. We also devised different methods to evaluate the level of privacy risk determined by the publication of personal images on social networks. We carried out several experiments to evaluate the effectiveness of our methods. Results show that it is feasible to reliably determine the kind of data involved in privacy policies based on their natural language description. With our technique for evaluating the privacy level of personal images, we also obtained a significant correlation between our algorithm's predictions and the ground truth. We also devised and developed a preliminary prototype of an integrated system for helping the user in taking informed decisions regarding the online sharing of personal contents, which includes a chatbot for natural language communication.

This work can be extended and improved in different directions. First of all, we plan to extend and refine the integrated system by interacting with groups of users for improving its capabilities and natural language communication skills. We will also experiment with the preliminary prototype with a groups of users to evaluate utility and usability. We are considering extending the framework to support other actions, such as the release of video or textual data on social networks. Another interesting research direction is to personalize the technique for suggesting the privacy level, exploiting the history of privacy choices of the user using an active learning approach. The technique for determining the image privacy level could be improved by considering additional features extracted from sophisticated computer vision tools. For information retrieval from privacy policies, we will investigate machine learning methods to deal with imbalanced datasets, such as resampling methods. Finally, we plan to improve the chatbot adopting advanced methods for natural language processing and generation.

Author Contributions: Conceptualization, F.C., A.D., S.D., M.M. and D.R.; methodology, F.C., A.D., S.D., M.M. and D.R.; software, F.C., A.D., S.D., M.M. and D.R.; validation, F.C., A.D., S.D., M.M. and D.R.; formal analysis, F.C., A.D., S.D., M.M. and D.R.; investigation, F.C., A.D., S.D., M.M. and D.R.; resources, F.C., A.D., S.D., M.M. and D.R.; writing–original draft preparation, F.C., A.D., S.D., M.M. and D.R.; writing–review and editing, F.C., A.D., S.D., M.M. and D.R.; visualization, F.C., A.D., S.D., M.M. and D.R.; supervision, D.R.; project administration, D.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All data used in our experiments are available online.

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

References

1. Shklovski, I.; Mainwaring, S.D.; Skúladóttir, H.H.; Borgthorsson, H. Leakiness and creepiness in app space: Perceptions of privacy and mobile app use. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Toronto, ON, Canada, 26 April–1 May 2014; pp. 2347–2356.
2. Acquisti, A.; Brandimarte, L.; Loewenstein, G. Privacy and human behavior in the age of information. *Science* **2015**, *347*, 509–514. [CrossRef] [PubMed]
3. Choi, H.; Chakraborty, S.; Charbiwala, Z.M.; Srivastava, M.B. Sensorsafe: A framework for privacy-preserving management of personal sensory information. In *Workshop on Secure Data Management*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 85–100.
4. Kim, S.H.; Ko, H.G.; Ko, I.Y.; Choi, D. Effects of contextual properties on users' privacy preferences in mobile computing environments. In Proceedings of the 2015 IEEE Trustcom/BigDataSE/ISPA, Helsinki, Finland, 20–22 August 2015; Volume 1, pp. 507–514.
5. Brodie, C.; Karat, C.; Karat, J. An empirical study of natural language parsing of privacy policy rules using the SPARCLE policy workbench. In Proceedings of the Second Symposium on Usable Privacy and Security, Pittsburgh, PA, USA, 12–14 July 2006; Volume 149, pp. 8–19.
6. Bhatia, J.; Breaux, T.D. Towards an information type lexicon for privacy policies. In Proceedings of the Eighth IEEE International Workshop on Requirements Engineering and Law, RELAW 2015, Ottawa, ON, Canada, 25 August 2015.
7. Zimmeck, S.; Bellovin, S.M. Privee: An Architecture for Automatically Analyzing Web Privacy Policies. In Proceedings of the USENIX Security Symposium, San Diego, CA, USA, 20–22 August 2014.
8. Narouei, M.; Takabi, H.; Nielsen, R. Automatic extraction of access control policies from natural language documents. *IEEE Trans. Dependable Secur. Comput.* **2018**, *17*, 506–517. [CrossRef]
9. Spruit, M.; Ferati, D. Text Mining Business Policy Documents: Applied Data Science in Finance. *Int. J. Bus. Intell. Res. IJBIR* **2020**, *11*. [CrossRef]
10. Zimmeck, S.; Story, P.; Smullen, D.; Ravichander, A.; Wang, Z.; Reidenberg, J.R.; Russell, N.C.; Sadeh, N. Maps: Scaling privacy compliance analysis to a million apps. *Proc. Priv. Enhancing Technol.* **2019**, *2019*, 66. [CrossRef]
11. Fan, M.; Yu, L.; Chen, S.; Zhou, H.; Luo, X.; Li, S.; Liu, Y.; Liu, J.; Liu, T. An Empirical Evaluation of GDPR Compliance Violations in Android mHealth Apps. In Proceedings of the 2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE), Coimbra, Portugal, 12–15 October 2020; pp. 253–264.
12. Zerr, S.; Siersdorfer, S.; Hare, J.S.; Demidova, E. Privacy-aware image classification and search. In Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, Portland, RO, USA, 12–16 August 2012; pp. 35–44.
13. Viola, P.A.; Jones, M.J. Robust Real-Time Face Detection. *Int. J. Comput. Vis.* **2004**, *57*, 137–154. [CrossRef]
14. Squicciarini, A.C.; Caragea, C.; Balakavi, R. Analyzing images' privacy for the modern web. In Proceedings of the 25th ACM Conference on Hypertext and Social Media, HT '14, Santiago, Chile, 1–4 September 2014; pp. 136–147.
15. Vailaya, A.; Jain, A.K.; Zhang, H. On image classification: City images vs. landscapes. *Pattern Recognit.* **1998**, *31*, 1921–1935. [CrossRef]
16. Kim, E.; Helal, S.; Cook, D. Human Activity Recognition and Pattern Discovery. *IEEE Pervasive Comput.* **2010**, *9*, 48–53. [CrossRef]
17. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
18. Yang, J.; Jiang, Y.; Hauptmann, A.G.; Ngo, C. Evaluating bag-of-visual-words representations in scene classification. In Proceedings of the 9th ACM SIGMM International Workshop on Multimedia Information Retrieval, MIR 2007, Augsburg, Bavaria, Germany, 24–29 September 2007; pp. 197–206.
19. Tonge, A.; Caragea, C. Image privacy prediction using deep neural networks. *ACM Trans. Web* **2020**, *14*, 1–32. [CrossRef]
20. Yu, J.; Kuang, Z.; Zhang, B.; Zhang, W.; Lin, D.; Fan, J. Leveraging content sensitiveness and user trustworthiness to recommend fine-grained privacy settings for social image sharing. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1317–1332. [CrossRef]
21. Google. Google Privacy Policy. 2019. Available online: https://policies.google.com/privacy?hl=en-US (accessed on 20 May 2021).
22. Siddula, M.; Li, L.; Li, Y. An empirical study on the privacy preservation of online social networks. *IEEE Access* **2018**, *6*, 19912–19922. [CrossRef]

23. Altınel, B.; Ganiz, M.C. Semantic text classification: A survey of past and recent advances. *Inf. Process. Manag.* **2018**, *54*, 1129–1153. [CrossRef]
24. Natural Language Toolkit. 2019. Available online: https://www.nltk.org/ (accessed on 20 May 2021).
25. Dekel, O.; Shamir, O. Multiclass-multilabel classification with more classes than examples. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, Sardinia, Italy, 13–15 May 2010; pp. 137–144.
26. Read, J.; Pfahringer, B.; Holmes, G.; Frank, E. Classifier chains for multi-label classification. *Mach. Learn.* **2011**, *85*, 333. [CrossRef]
27. Parmar, A.; Katariya, R.; Patel, V. A Review on Random Forest: An Ensemble Classifier. Available online: https://www.researchgate.net/publication/329820638_A_Review_on_Random_Forest_An_Ensemble_Classifier (accessed on 20 May 2021).
28. Wang, Z.; Cui, P.; Li, F.; Chang, E.Y.; Yang, S. A data-driven study of image feature extraction and fusion. *Inf. Sci.* **2014**, *281*, 536–558. [CrossRef]
29. Montazer, G.A.; Giveki, D. Content based image retrieval system using clustered scale invariant feature transforms. *Optik* **2015**, *126*, 1695–1699. [CrossRef]
30. Squicciarini, A.C.; Novelli, A.; Lin, D.; Caragea, C.; Zhong, H. From Tag to Protect: A Tag-Driven Policy Recommender System for Image Sharing. In Proceedings of the 15th Annual Conference on Privacy, Security and Trust, PST 2017, Calgary, AB, Canada, 28–30 August 2017; pp. 337–348.
31. Gao, S.; Wang, Z.; Chia, L.; Tsang, I.W. Automatic image tagging via category label and web data. In Proceedings of the 18th International Conference on Multimedia 2010, Firenze, Italy, 25–29 October 2010; pp. 1115–1118.
32. De Meo, P.; Ferrara, E.; Fiumara, G.; Provetti, A. Generalized Louvain Method for Community Detection in Large Networks. In Proceedings of the 2011 11th International Conference on Intelligent Systems Design and Applications, Cordoba, Spain, 22–24 November 2011; pp. 88–93. [CrossRef]
33. Frank, E.; Hall, M.A.; Holmes, G.; Kirkby, R.; Pfahringer, B. WEKA—A Machine Learning Workbench for Data Mining. In *The Data Mining and Knowledge Discovery Handbook*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 1305–1314.