**International Journal of Information Security and Privacy**
Volume 15 • Issue 1 • January-March 2021

# Utilizing Sentence Embedding for Dangerous Permissions Detection in Android Apps' Privacy Policies

Rawan Baalous, University of Glasgow, UK
Ronald Poet, University of Glasgow, UK

## ABSTRACT

Privacy policies analysis relies on understanding sentences meaning in order to identify sentences of interest to privacy related applications. In this paper, the authors investigate the strengths and limitations of sentence embeddings to detect dangerous permissions in Android apps privacy policies. Sent2Vec sentence embedding model was utilized and trained on 130,000 Android apps privacy policies. The terminology extracted by the sentence embedding model was then compared with the gold standard on a dataset of 564 privacy policies. This work seeks to provide answers to researchers and developers interested in extracting privacy related information from privacy policies using sentence embedding models. In addition, it may help regulators interested in deploying sentence embedding models to check for privacy policies' compliance with the government regulations and to identify points of inconsistencies or violations.

## KEYWORDS

Android Permissions, Privacy Policies, Run Time Permissions, Sentence Embedding

## INTRODUCTION

Android apps may collect, use and share users' personal information for several purposes. To support users' privacy, Google requires apps which access users' personal information to post a privacy policy which discloses how the app handles users' information and for what purposes (Google, 2017). Such policies are also intended to fulfill legal requirements by the law to protect users' privacy (Wang et al., 2019). Privacy policies support users in privacy making decisions by answering questions such as: what information will be collected from users? what the collected information will be used for? which parties will the information be shared with? For how long the information will be stored? And so on. When users accept the privacy policy, this means that they agree to release their data under the conditions specified by the privacy policy (Costante, Sun, Petković, & den Hartog, 2012).

Although privacy policies are the main source of companies' data handling practices, most users do not read privacy policies before using the services (Furnell, & Phippen, 2012). There seems to be contradictory results between studies showing users' concerns about their privacy, and that they often don't read privacy policies. One possible explanation could be related to the complexity of reading policies. Although users would like to protect their privacy in principle, they feel that this is a difficult task in practice. Hence, they give up trying to preserve control over their privacy. In addition, actually reading all encountered privacy policies looks like an impossible task (Steinfeld, 2016).

DOI: 10.4018/IJISP.2021010109

Automatic analysis of privacy policy documents may have a great advantage on extracting specific privacy information related to users' queries. However, privacy policies automatic analysis relies on understanding sentences meaning in order to identify sentences of interest to users' queries or privacy related applications. Moreover, privacy policies are often written in natural language, and hence use a wide range of expressions to describe the information types they collect, use, and share. In contrast, Android Application Program Interface (API) methods use limited terminology to describe the collected users' personal information (Hosseini, Qin, Wang, & Niu, 2018).

Figure 1 illustrates the variability of natural language expressions in Android apps' privacy policies. In the first sentence, the dangerous permissions that allow the app to access user's (phone number) and (address) are combined into a more generalized data type (contact information). In the second sentence, the same data type (contact information) is used to denote the ability of the app to access user's (address book), which is another dangerous permission. In the last sentence, the data type (contact information) encompasses the same two dangerous permissions in the first sentence, in addition to a third dangerous permission which allows the app to access user's (accounts) on the phone, such as social networking accounts. Privacy policies commonly use hypernym relation (a more general phrase that has sub ordinates) to describe their data practices (Bhatia, Evans, Wadkar, & Breaux, 2016). Using this relation throughout the privacy policy can cause multiple interpretations of the same data practice.

Back in 1955, the project on artificial intelligence (AI) was introduced by the assumption that aspects of learning or features of intelligence can be so precisely described so that machines can simulate them (McCarthy, Minsky, Rochester, & Shannon, 2006). Afterward, several efforts have been made to improve machines to be able to work just like human and solve complex problems. A fundamental aspect of being human is the capability of comparing things and discovering their relatedness. In this regard, various machine learning models were developed to compare semantic entities such as words and sentences (Harispe, Ranwez, Janaqi, & Montmain, 2015).

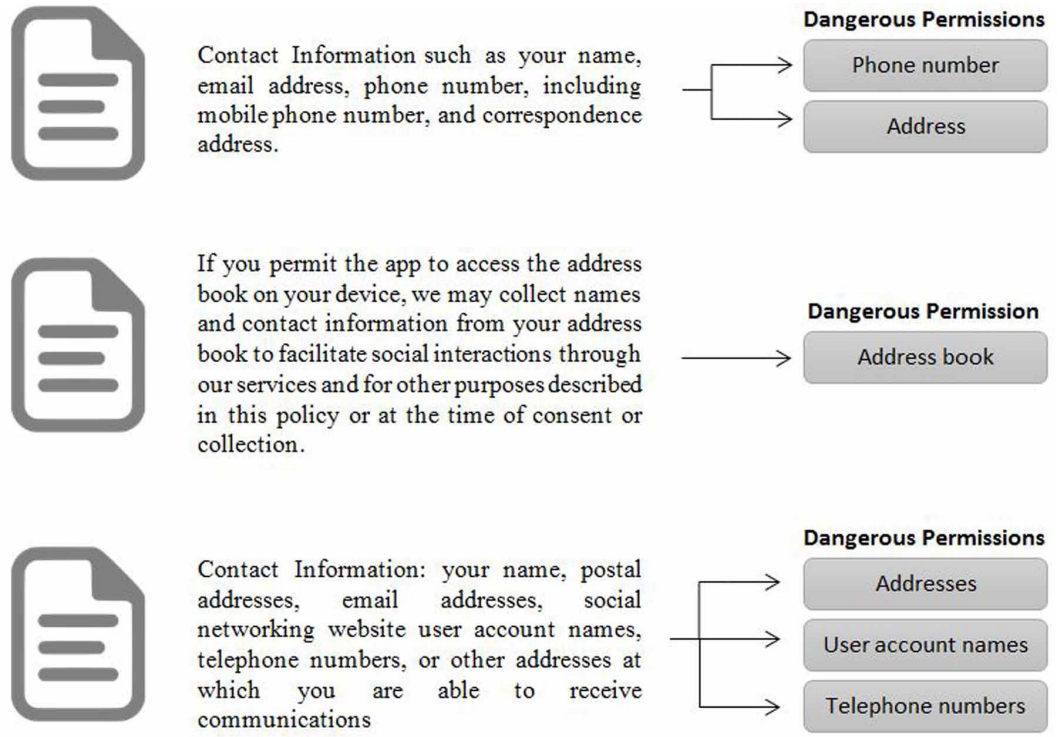
Sentence embedding models are promising techniques that are used to capture sentences semantics and their relations. There are different applications that rely on encoding semantic meaning of privacy policies sentences, such as applications interested in checking Android apps' behaviors against what is stated in their privacy policies. In fact, many privacy related applications such as Liu, Fella, & Liao (2016), Gopinath, Wilson, & Sadeh (2018), Sun (2018), and Harkous et al. (2018) use word or sentence embedding models as part of the automatic analysis of privacy policies. However, it is not entirely clear to what extent sentence embeddings are effective in capturing the semantics of privacy policies sentences. Therefore, to ensure the successfulness of such applications, it is crucial to report the advantages and disadvantages of using sentence emebddings and suggest improvements if needed.

In this work, first, we created a taxonomy for Android dangerous permissions' terminology in privacy policies, on a dataset of 564 Android apps. We considered Android dangerous permissions only since these permissions involve users' private information and require users' approval before granting (Android Developers, 2018). The taxonomy provides the ground truth data for evaluating the effectiveness of the sentence embedding model. Second, we made an extensive comparison between the dangerous permissions actually included in Android apps' privacy policies (gold standard) with the dangerous permissions extracted by the sentence embedding model. To the best of our knowledge, no work has been done in evaluating the effectiveness of sentence embeddings in privacy policy documents.

## ANDROID DANGEROUS PERMISSIONS

Prior to Android Marshmallow, when a user starts the process of installing an Android app, he will be presented with a list of permissions that the app requests. This permissions' screen names all the phone resources that the app will access, if granted. For example, an app requesting the permission SEND_SMS will be able to send SMS messages, if installed, but an app without this permission

**Figure 1. Variability of natural language expressions in Android apps' privacy policies**




cannot. The user has to decide either to accept all the requested permissions and install the app, or cancel the installation process. Users are not shown permissions at any time other than installation (Felt et al., 2012).

After several years of using this approach, Android started a new model in October 5, 2015, with the code name Marshmallow. Since Marshmallow, the installation of the app is completed irrespective of the required permissions. After that, each time the user starts an action in the application which requires dangerous permission, a pop-up window appears to the user asking to grant the permission. Consequently, the user has much more control since he can grant and deny permissions individually. Furthermore, the user can also revoke permissions later after the installation (Alepis, & Patsakis, 2017).

Some permissions in Android are considered normal permissions, which means that there is no high risk to the user's privacy in allowing apps accessing them. In this case, these permissions are automatically granted at install time. In addition, users are unable to revoke them. Allowing the app to set the time zone or vibrate the device are examples of normal permissions. On the other hand, dangerous permissions are permissions that can affect users' privacy or the operation of other apps, and must be granted explicitly by the users. For example, the permission associated with reading the user's contacts is considered a dangerous permission (Android Developers, 2018). Dangerous permissions are categorized into groups, as illustrated in Table 1.

## SENTENCE EMBEDDING

Sentences embeddings are used to find out the similarity scores between sentences, which can be used later in further Natural Language Processing (NLP) tasks. However, only few works have been done on learning sentence embeddings that can be used easily and effectively across several domains in the

**Table 1. Android dangerous permission groups (Android Developers, 2018)**

| Permission Group | Permissions |
|---|---|
| **Calendar** | READ_CALENDAR<br>WRITE_CALENDAR |
| **Camera** | CAMERA |
| **Contacts** | READ_CONTACTS<br>WRITE_CONTACTS<br>GET_ACCOUNTS |
| **Location** | ACCESS_FINE_LOCATION<br>ACCESS_COARSE_LOCATION |
| **Microphone** | RECORD_AUDIO |
| **Phone** | READ_PHONE_STATE<br>READ_PHONE_NUMBERS<br>CALL_PHONE<br>ANSWER_PHONE_CALLS<br>READ_CALL_LOG<br>WRITE_CALL_LOG<br>ADD_VOICEMAIL<br>USE_SIP<br>PROCESS_OUTGOING_CALLS |
| **Sensors** | BODY_SENSORS |
| **SMS** | SEND_SMS<br>RECEIVE_SMS<br>READ_SMS<br>RECEIVE_WAP_PUSH<br>RECEIVE_MMS |
| **Storage** | READ_EXTERNAL_STORAGE<br>WRITE_EXTERNAL_STORAGE |

same sense as word embeddings (Wieting, Bansal, Gimpel, & Livescu, 2015). The approaches used to model sentences range from simple ones that are based, for example, on word averaging (Arora, Liang, & Ma, 2016) to more complex neural network architectures (Tai, Socher, & Manning, 2015). While deep learning approaches are powerful and very strong in expressiveness, the complexity in such models makes them slower to train on larger datasets. On the other hand, simpler models such as matrix factorizations take advantage of training on larger datasets (Pagliardini, Gupta, & Jaggi, 2017).

Recently, a sentence embedding model, Sent2Vec (Pagliardini, Gupta, & Jaggi, 2017) significantly outperformed the state-of-the-art unsupervised sentence embedding models on most benchmark tasks. In contrast to neural network based architectures, the model is simple, and the complexity of training as well as inference is low. Hence, the model can be trained on very large datasets in a short amount of time. The proposed model can be considered as an extension of the Continuous Bag-of-Words (CBOW) architecture (Mikolov, Chen, Corrado, & Dean, 2013). Sent2Vec computes sentence embedding using word vectors along with n-gram embeddings.

## RELATED WORK

There has been an increased interest in applying natural language processing techniques and machine learning methods to privacy policies, in order to improve their effectiveness (Lebanoff, & Liu, 2018). For example, a machine learning solution was proposed by Costante, Sun, Petković, & den Hartog (2012) to automatically evaluate privacy policy's completeness. The authors extracted privacy

categories from privacy regulations and used machine learning methods to find out the categories that are covered by the privacy policies. Another study by Liu, Fella, & Liao (2016) used neural networks to model the vagueness of privacy policies. Recently, a detailed analysis on privacy policies (Harkous et al., 2018) used NLP and deep learning to automatically analyze privacy policies. The researchers proposed an application which can answer users' queries on natural language privacy policies.

Much recent work has focused on comparing Android permissions with natural language description of Android apps. In WHYPER (Pandita, Xiao, Yang, Enck, & Xie, 2013) for example, the authors identified consistency between Android app's description and declared permissions. They extracted semantic patterns from Android API documents with the assumption that these keywords or semantic patterns can be found in the app's description, and are adequate in representing Android permissions. AutoCog (Qu et al., 2014) was then proposed to overcome the limitations of WHYPER, since Android API documents are limited in the amount of semantic patterns that can be correlated with Android permissions. Therefore, AutoCog relies on extracting semantic patterns from app's description and shows related parts in the app's description which implies the permissions. Close to WHYPER and AutoCog, Feng, Chen, Zheng, Gao, & Zheng (2019) in AC-Net tackled the same problem but differently. Instead of outputting a "Yes" or "No" answer to the consistency between app's description and permissions, AC-Net provides the degree of consistency using deep learning techniques. Olukoya, Mackenzie, & Omoronyia (2020) on the other hand, investigated the consistency between app's description and permissions with the aim of improving malware detection. Their proposed technique could detect malwares with a precision of 90%.

Other works were interested in processing Android apps' privacy policies to check for privacy compliance with actual apps' behavior. For example, Slavin et al. (2016) assessed consistency between Android apps' privacy policies and apps' code. The semi-automated framework links privacy policy phrases to API methods which produce sensitive information. Information flow analysis was then used to check if the collected data is sent outside Android apps on a dataset of 477 Android apps. Two years later, Wang et al. (2018) proposed an approach to detect inconsistencies between Android apps' privacy policy collection statements, apps code and collection behavior of user entered data through graphical user interface (GUI). The approach was evaluated on a dataset of 120 Android apps collected from three categories in Google Play Store. The results demonstrated some privacy leakage which violates Android apps' privacy policies.

Many approaches use word or sentence embedding models as part of the automatic analysis of privacy policies, such as Liu, Fella, & Liao (2016), Gopinath, Wilson, & Sadeh (2018), Sun (2018), and Harkous et al. (2018). However, as mentioned earlier, it is not entirely clear to what extent these embeddings are effective in capturing the semantics of privacy policies. This work is intended to fill this gap by examining the effectiveness of sentence emebeddings in extracting privacy related information from privacy policies. Most of the studies evaluate word or sentence embeddings in general English text (Schnabel, Labutov, Mimno, & Joachims, 2015) and (Perone, Silveira, & Paula, 2018). There has been some works to evaluate word or sentence emebeddings in specific domains, such as psychology (Altszyler, Ribeiro, Sigman, & Slezak, 2017), biomedical (Chen, Peng, & Lu, 2019), and geoscience (Padarian, & Fuentes, 2019). However, as pointed out in the literature (Lu, Yu, Shi, & Li, 2018), most of the work done in many NLP tasks covers general domain texts. The lack of a training data set as well as evaluation data set limited the number of research in specific domains. Nevertheless, domain-specific applications are extensively needed these days.

## METHODOLOGY

### Dataset

We chose the top 641 apps from Google Play Store. The apps were chosen from all Google Play Store categories (entertainment, finance, games … etc.). The privacy policies of these apps were downloaded on October 2018. Since some apps don't have privacy policies and some provided incorrect or broken

links to their privacy policies, we ended up with 564 privacy policies. The analyzed privacy policies contain a total of 1,582,403 words.

## Gold Standard

To construct the gold standard, we followed the method of Baalous & Poet (2018), in which NLP and information extraction (IE) techniques were used to extract types of information that are collected, shared, used, or retained from our dataset of 564 Android apps' privacy policies. Then, the extracted types of information were categorized using synonym, hypernym and meronym relationships with Android dangerous permissions.

## Semantic Sentence Embedding

In this section, we detail our approach to mine Android dangerous permissions from Android apps' privacy policies using semantic sentence embedding. The process is composed of extracting noun phrases representing dangerous permissions, training the sentence embedding model, finding semantically related phrases to the dangerous permissions, and comparing the results generated by the sentence embedding model with the gold standard.

Briefly, Google provides list of dangerous permissions that affect users' privacy (Android Developers, 2018). The dangerous permissions are organized into groups and have specific syntax, such as (READ_EXTERNAL_STORAGE) and (WRITE_EXTERNAL_STORAGE) which belong to the storage group. Given this list of dangerous permissions, first we extracted noun phrases representing dangerous permissions (e.g. extract noun phrase "external storage" from "READ_EXTERNAL_STORAGE" and "WRITE_EXTERNAL_STORAGE" permissions). Second, we trained the machine learning model with 130,000 Android apps' privacy policies. Third, we used the trained model to find semantically related phrases to the dangerous permissions on the test set of 564 privacy policies. Finally, in order to evaluate the performance of the machine learning model, we compared the generated phrases that are semantically related to the dangerous permissions with the gold standard. Figure 2 presents our approach.

### *Extract Noun Phrases Representing Dangerous Permissions*

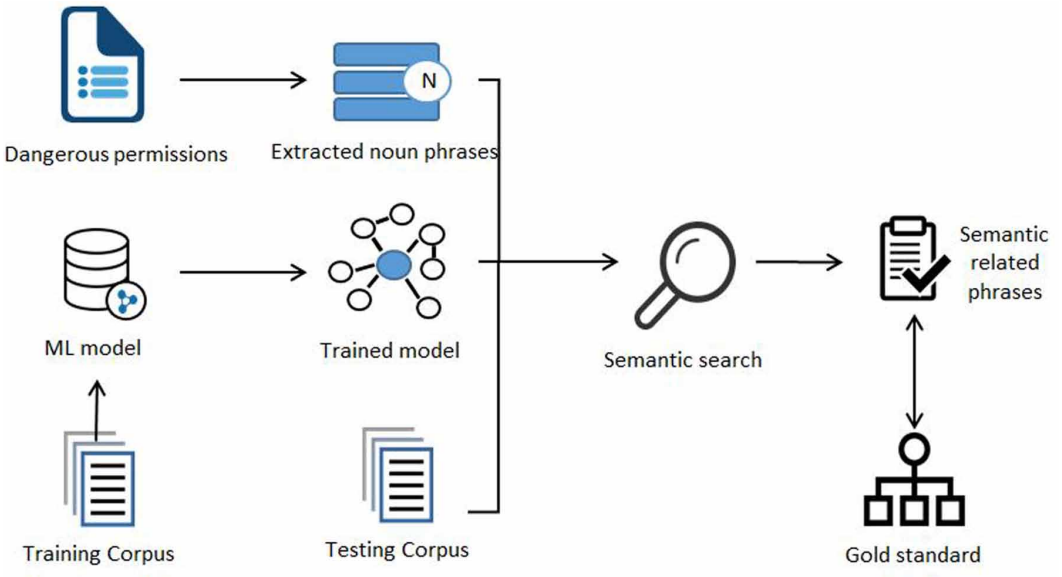
Note that there are a couple of preprocessing steps that were applied to the list of Android dangerous permissions using the NLTK library. These steps prepared dangerous permissions for extracting noun phrases that will be fed later into the machine learning model. First, we replaced all underscores with white spaces. Then, we converted all words to lowercase. Third, we broke each permission down into its words and attached a part of speech (POS) tag to each word. In this step, we chained different taggers together to increase accuracy, so that if one tagger failed to tag a word, the word is passed to the next backoff tagger, and so on. We manually check that the tags are correct afterwards.

In order to extract noun phrases, we followed the method of Kim, Baldwin, & Kan (2010) in which a set of regular expression grammars corresponding to noun phrases (NP) are defined and parsed. After that, we removed duplicate noun phrases. Finally, we manually reviewed the resulted list of noun phrases representing dangerous permissions. Figure 3 provides an example of extracting noun phrases representing the (READ_CONTACTS) and (WRITE_CONTACTS) dangerous permissions.

### *Train The Machine Learning Model*

Prior to finding semantic related phrases, the machine learning model has to be trained. While a general purpose dataset such as Google News or Wikipedia with pre-trained models are publically available, it has been shown that the word embedding model trained on a small domain specific dataset can outperform the word embedding model trained on a large generic dataset (Sugathadasa et al., 2017). Thus, we have considered dataset provided by (Harkous et al., 2018) for training the machine learning model. The chosen dataset has several advantages: First, it contains a large number

**Figure 2. Using machine learning model to mine dangerous permissions from Android apps' privacy policies**




of privacy policies (130,000). Second, the privacy policies were retrieved from Google Play Store, hence the content of the privacy policies reflect the data collected by Android apps.

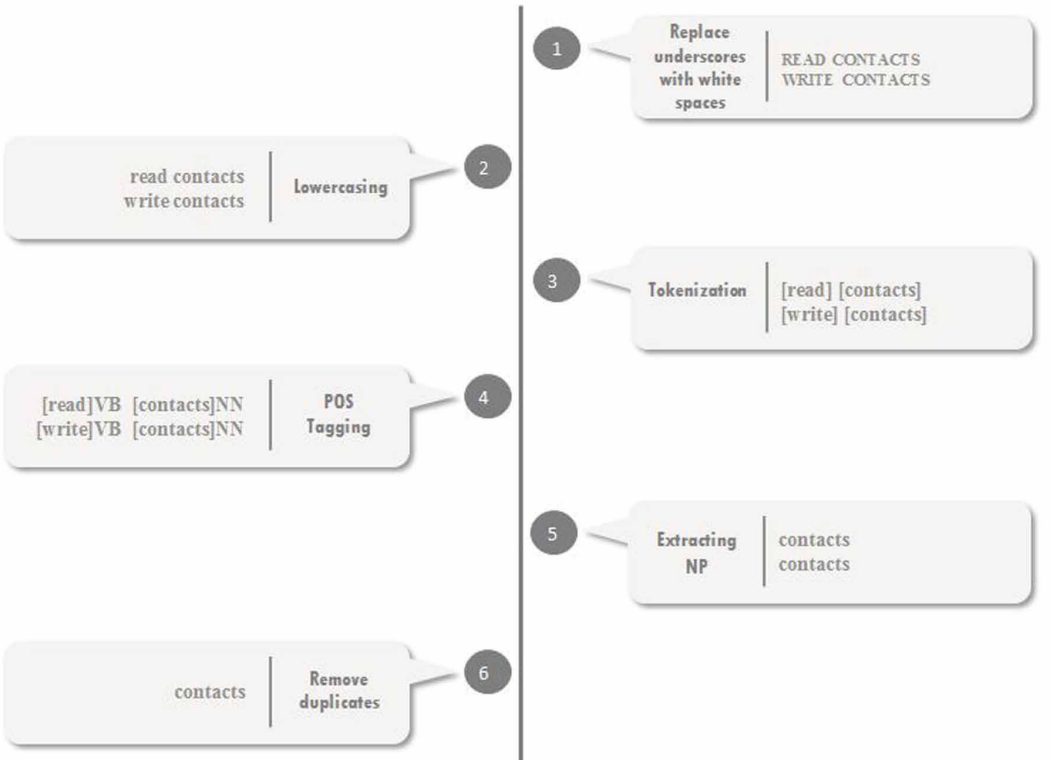
We used the Sent2Vec model (Pagliardini, Gupta, & Jaggi, 2017) in our study since it has been shown that Sent2Vec significantly outperforms the state of the art unsupervised models for most benchmark tasks. In contrast to other sentence embedding models such as SkipThought, the simplicity of Sent2Vec makes the computational cost low for training as well as inference. This allows the model to quickly learn from large datasets.

To train the Sent2Vec model, the training privacy policies text file must contain one sentence per line. Hence, we performed additional pre-processing steps to prepare the 130,000 privacy policies text for training. First of all, we segmented the text file into one sentence per line. Next, we lowercased all the words and tokenized the text. Finally, we filtered out very long sentences (e.g. privacy policy that is retrieved as one long sentence) to avoid the model crashing in training. The model was trained on approximately 3.9 million sentences, containing 110 million words.

### *Find Semantic Related Phrases*

We utilized the Sent2Vec trained model to find most similar phrases to each extracted noun phrase representing dangerous permission. In order to do so, we tested the Sent2Vec model on the same privacy policies dataset used to construct the gold standard. This dataset was used to find ground truth terminology representing dangerous permissions and will be compared later to the phrases extracted by the Sent2Vec model. The nearest neighboring phrases were extracted by the Sent2Vec model according to the cosine distance, to find the closest vector representations to the dangerous permissions. The cosine similarity score ranges between 0 (which means that there is no similarity between compared phrases) and 1 (which means that the compared phrases are absolutely similar). We filtered the top similar phrases for each dangerous permission based on the following: if the cosine similarity between the generated phrase and the dangerous permission is >= 0.5, then we accept it, otherwise we reject it. The chosen threshold is based on prior work on measuring semantic similarity of text (Mihalcea, Corley, & Strapparava, 2006), (Abdelali, Cowie, & Soliman, 2007), (Al-Kouz,

**Figure 3. Extract noun phrases representing dangerous permissions**




Luca, & Albayrak, 2011), (Crocetti, 2015), and (Kamaruddin, Yusof, Bakar, Tayie, &Alkubaisi, 2018). Here are some examples of the semantic related phrases extracted by the Sent2Vec model:

1. **Approximate location** (network-based) this permission allows northeastappsinc to identify and display your location on map or apps installed by anonymous surrounding users and to recommend popular apps based on users' location.
2. This application may send **push notifications** to the user.

### *Compare Results with the Gold Standard*

As discussed earlier, in the gold standard we extracted the terminology used in privacy policies that are relevant to dangerous permissions. For example, the terminology (approximate location) and (imprecise geolocation) in Android apps' privacy policies were mapped to (ACCESS_COARSE_LOCATION) dangerous permission, while the terminology (precise location) and (exact geo-coordinates) were mapped to (ACCESS_FINE_LOCATION) dangerous permission. In this section, we compared the results generated by the machine learning model with the phrases recorded in the gold standard. For example, as can be seen from the semantic related phrases extracted by the Sent2Vec model in the previous section, the model considered the first sentence to be semantically related to (ACCESS_FINE_LOCATION) dangerous permission. However, the sentence talks about the network based location instead, and contains the terminology (approximate location) which is mapped to (ACCESS_COARSE_LOCATION) dangerous permission in the gold standard. Hence, this sentence is considered false positive. More detail on the difference between precise location (ACCESS_FINE_LOCATION) and approximate location (ACCESS_COARSE_LOCATION) can

be found in the discussion section. On the other hand, the model considered the second sentence to be semantically related to (RECEIVE_WAP_PUSH) dangerous permission. This is considered true positive as (push notifications) terminology in this sentence is mapped to (RECEIVE_WAP_PUSH) dangerous permission in the gold standard.

## RESULTS

To ensure that the sentence embedding model was indeed selecting relevant phrases to Android dangerous permissions, we calculated the precision, recall and F1 against the gold standard. Table 2 shows the performance of the sentence embedding model. The highest result achieved in each metric is marked with bold face. As presented in the table, "Calendar" and "Sensors" Android dangerous permission groups have no results, since there were no semantic related phrases selected by the sentence embedding model with cosine similarity equal or superior to 0.5.

Across all Android dangerous permission groups, the average recall rate was 0.18, which demonstrates that many relevant phrases to Android dangerous permissions were not extracted by the sentence embedding model. On the other hand, the average precision rate was 0.84, which shows that the phrases selected by the sentence embedding model contain only a few errors, and most of the phrases extracted are relevant to Android dangerous permissions. However, the large number of unselected phrases related to dangerous permissions significantly affected the recall rate, and hence the F1 value.

There could be several reasons behind the low F1 value. First, many terminology included in the gold standard appeared only once in Android apps' privacy policies. These low frequency words were not detected by the sentence embedding model. In the training phase, the (min_count) parameter was set to 5, which means that the model will ignore all words with total frequency lower than 5. This configuration will speed up the training time especially that many privacy policies are very long documents with thousands of words. It will also decrease the amount of memory needed for embeddings. On the other hand, it will affect the vocabulary size. Another possible reason is the number of privacy policies documents used for training the model. A larger number may increase the total number of the vocabulary.

In all dangerous permission groups, the lowest F1 score was reached with the (Contacts) permission group. This dangerous permission group contains three permissions: (READ_CONTACTS), (WRITE_CONTACTS) and (GET_ACCOUNTS). Granting this dangerous permission group will allow the app to read, edit and add contacts as well as get access to the accounts the user use in his device, such as Twitter and Facebook accounts. When using the sentence embedding model to find the nearest neighbors to the (READ_CONTACTS) and (WRITE_CONTACTS) for example, the semantic related sentences contain only the word (contacts), while the gold standard contains several other semantically related terminology such as (address book) and (phone book).

The most frequent relation type extracted by the sentence embedding model is synonym, in which the meaning of the dangerous permission and the selected phrase by the model are equivalent for our purposes. On the other hand, other types of relations such as meronym (part-to-whole relationship) were never detected by the model. For example, granting the (Location) dangerous permission group will allow the Android app to determine the user's approximate and/or precise location. The terminology: (location data) and (geographic location) selected by the sentence embedding model are all synonyms. However, there are other terminology in the gold standard such as (country), (city), (town) and (state) which are all considered part of the location (meronym relation), were not detected by the sentence embedding model. Furthermore, the sentence embedding model detected few terminology correspond to hypernym relationship. For example, (push notifications) was selected by the sentence embedding model as it is semantically related to (RECEIVE_WAP_PUSH) dangerous permission. This relationship is considered a hypernym, since (push notifications) has a broad meaning and includes Short Messaging Service (SMS-PUSH), Multimedia Messages Service (MMS-PUSH),

**Table 2. Results from the sentence embedding model compared against the gold standard**

| Permission Group | Precision | Recall | F1 |
|---|---|---|---|
| **Calendar** | - | - | - |
| **Camera** | 0.80 | **0.66** | **0.72** |
| **Contacts** | 0.80 | 0.04 | 0.07 |
| **Location** | 0.72 | 0.11 | 0.19 |
| **Microphone** | 0.90 | 0.15 | 0.25 |
| **Phone** | 0.70 | 0.15 | 0.24 |
| **Sensors** | - | - | - |
| **SMS** | **1** | 0.11 | 0.19 |
| **Storage** | **1** | 0.08 | 0.14 |

Wireless Application Protocol (WAP-PUSH), etc., (Guo, & Liu, 2013). However, this hypernym relationship was rarely detected by the sentence embedding model.

## DISCUSSION

Information extraction is an in depth understanding task. It requires the relevant content to the user's need to be located and extracted from the document (Turmo, Ageno, & Català, 2006). Users usually require an answer of a specific question, for example "Why does this app collect my location?". In order for the user to answer this question, he needs to spend time and do extra work looking for sentences in the privacy policy talking about location. Note that privacy policies may use other semantically related phrases to location which also make the extraction task more difficult. Sentence embedding models are promising techniques for simplifying information extraction procedure and improving extraction performance.

Users can take advantage of sentence embedding models to search for specific data practices that might be hidden in long privacy policy documents, such as third party tracking or marketing advertisements. A user might type for example "we use your location for marketing advertisements" and the sentence embedding model will extract the relevant privacy policies sentences. The output can then be used to make more privacy informed decisions when deciding to allow or deny Android dangerous permissions requests. In addition, sentence embedding models might help regulators in checking for privacy compliance. According to (Harkous et al. 2018), various studies conducted by regulators analyzed manually the permissiveness of compliance checks. Furthermore, the number of investigated privacy policies is usually in the range of tens. In 2000, the Federal Trade Commission (FTC) conducted a survey on a collection of U.S. websites to examine their compliance with four fair information practice principles: access, security, notice and choice. Commission staffs reviewed all the printed privacy documents and answered some questions regarding their content (Federal Trade Commission, 2000). Sentence embedding models can play an important rule here in matching regulators' queries with answers from privacy policy documents in a very short time. Regulators can build on the output to check for privacy compliance of privacy policies with data protection regulations, such as the General Data Protection Regulation (GDPR).

When we build IE systems, it is crucial to evaluate the systems so that we see how they behave with respect to golden standards. Depending on the IE system, certain performance measures might be considered more important than others. For instance, high precision results are greatly recommended when the IE system does not control the extraction results manually. On the other hand, if the IE system does the extraction automatically and the information extraction task is more of an initial

filtering in which a manual selection is performed afterward, then high recall results are greatly recommended (Moens, 2006).

The results presented in our work show high precision for using sentence embedding to extract dangerous permissions from Android apps' privacy policies. It appears that in most cases the model could accurately identify dangerous permissions. However, the ability of the model to find all relevant phrases to Android dangerous permissions is considered low (18%) compared to the actual number included in the privacy policies. In other words, if the privacy policy uses the terminology (USB storage) for example to describe accessing of external storage in Android phone, the sentence embedding model will fail to extract the relevant phrase.

The results further demonstrate that identifying phrases semantically related to dangerous permissions is not a trivial task. When constructing the gold standard, although we used a semi-automated approach, it took a massive amount of time to locate information types in each relevant sentence and also to manually map the extracted information type to Android dangerous permissions. Considering the vagueness of privacy policies, this makes the task more difficult and requires much more time to achieve it using a semi-automated method. In fact, it was reported that even privacy experts might not always agree on the interpretation of privacy policies (Reidenberg, 2015). On the other hand, using the sentence embedding model is a more straightforward and quick solution. However, given such ambiguity in privacy policies, we don't expect the sentence embedding model to perform perfectly.

With respect to precise and approximate location dangerous permissions (ACCESS_FINE_LOCATION and ACCESS_COARSE_LOCATION) respectively, the sentence embedding model extracted almost the same semantically related sentences for both permissions. Under the definition of dangerous permissions in Android, the two locations permissions are not considered similar. ACCESS_COARSE_LOCATION can't utilize the Global Positioning System (GPS). Instead, it employs Android's network location provider to get user's location through Wi Fi signals and cell towers. Therefore, this permission is used to acquire user's approximate location and is not as accurate as ACCESS_FINE_LOCATION (Android Developers, 2019). The sentence embedding model did not draw a finer line between both location permissions. It looks like it considered them identical. This can be attributed to the fact that both permissions contain the same word (location) and that they are very short sentences, which might make the distinction between them a challenging task. On the other hand, the gold standard data distinguishes between them. For example, (exact location), (precise location), and (exact geo-coordinates) are considered synonyms to ACCESS_FINE_LOCATION permission, while (non-precise geolocation), (approximate location) and (imprecise geolocation) are considered synonyms to ACCESS_COARSE_LOCATION permission.

Additional analysis on the nearest neighbors to dangerous permissions revealed the following: For dangerous permissions that contain abbreviations, such as (SMS), the sentence embedding model performed much better on the abbreviation of the dangerous permission compared to what the abbreviation stands for. To further clarify, when the dangerous permission input to the sentence embedding model is (SMS) for example, the generated nearest neighbors are more semantically related to the permission than when the input is the long form of the dangerous permission (short message service). This could be due to fact that such abbreviations are more commonly used by privacy policies than their long forms.

We also observed that many sentences are relevant to dangerous permissions although their cosine similarity is under 0.5. In fact, some dangerous permissions that have no nearest neighbors with a cosine score above the threshold 0.5, have some semantically similar sentences with cosine scores under 0.5. However, this observation can't be generalized to all dangerous permissions as the case differs from one dangerous permission to the other. This can be further tested by gradually decreasing the cosine similarity and observe the recall. Note that the recall can be improved by decreasing the threshold as the cosine similarity of many relevant sentences to dangerous permissions was under 0.5.

As mentioned earlier, the most detected relation by the sentence embedding model is synonym. Although few hypernym relations were returned by the model, no meronym relation was captured. These observations are in agreement with Handler (2014) in which the author reported that the Word2Vec embedding model captured certain relations ahead of others. For example, it was found that the Word2Vec favors synonyms ahead of meronyms. Since Sent2Vec sentence embedding model can be seen as an extension of Continuous Bag-of-Words (CBOW), which is one of the Word2Vec models (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013), and (Mikolov, Chen, Corrado, & Dean, 2013), there might be no surprise in the most detected relation types by Sent2Vec.

Finally, embedding sentences to vectors which preserve semantic meaning is a core step in many NLP tasks. To this end, different sentence embedding techniques have recently been proposed to compute sentence representations with impressive results (Voleti, Liss, & Berisha, 2019). However, real life tasks involve complicated forms of inference which makes it difficult for sentence embedding models to come to a conclusion (Conneau, Kruszewski, Lample, Barrault, & Baroni, 2018). In fact, this is truly presented in privacy policy sentences which use vague and broader meaning words to represent data handling practices. It could be thought that more complex sentence embedding models such as recurrent neural networks could provide promising results in this context. Nevertheless, this assumption needs to be tested in privacy policy sentences and compared to other sentence embedding models to shed light upon the advantages and disadvantages of each model.

## CONCLUSION

Extracting phrases semantically related to dangerous permissions from Android apps' privacy policies is a critical step in querying about dangerous permissions' collection and usage by Android apps. It could have several useful applications, such as mining the rationales of dangerous permissions from privacy policies, in order to assist users to make privacy informed decisions. In this work, we used the sentence embedding model Sent2Vec to find semantic related phrases to Android dangerous permissions from Android apps' privacy policies. For each dangerous permission, a list of closet neighbors was generated. The cosine similarity of 0.5 and greater was chosen as a threshold, which means that all vectors that are equal to 0.5 or greater are considered semantically close to the dangerous permissions. We compared the performance of Sent2Vec model with the gold standard and computed precision, recall and F1.

The results showed that the sentence embedding model was able to correctly capture semantically related phrases to Android dangerous permissions. The average precision rate was 0.84, which demonstrates that only few errors occurred. In contrast, the average recall rate was 0.18, which indicates that the generated list of semantically related phrases to Android dangerous permissions might not be adequate, as the model missed many related phrases in Android apps' privacy policies. Overall, the results suggest that we can get satisfactory results utilizing the semi-automated approach used to construct the gold standard. However, that method is time consuming and requires a lot of manual engineering. On the other hand, using a sentence embedding model is faster and cost effective, but the generated related phrases might not be sufficient.

As with all studies, the work in this paper is subject to some limitations. In order to filter the Android dangerous permissions similar phrases returned by the model, we accepted phrases with cosine similarity equal or superior to 0.5. The phrases with cosine similarity under 0.5 were rejected. We chose this threshold based on previous work on measuring semantic similarity of text. The major constrain here is the chosen cosine similarity threshold which affected the recall significantly. In fact, having Android privacy policies sample data first, testing empirically different cosine similarity thresholds, and choose the best value that produces more relevant results would provide stronger claim on the performance of the sentence embedding model. Another constraint is related to choosing the optimal training parameters. Minimum count parameter for example was set to 5, which means that the sentence embedding model will neglect all words that occur less than five times. Taking into

account the length of privacy policies documents, this configuration was chosen to speed up the training time and to decrease the amount of memory needed for embeddings. On the other hand, it reduced the vocabulary size generated by the model. Finally, having more test data in the evaluation data set would increase the usefulness of the study.

It would be interesting in the future to verify if the sentence embedding model can correctly select more dangerous permissions related phrases with gradually decreasing the threshold and observe the generated results. The variation of F measurement achieved with the changing of cosine similarity threshold can also be reported. Increasing the size of Android apps' privacy policies training data may also affect the overall results. It might be useful to test if the sentence embedding model will noticeably benefit from extra training data. Finally, future work may also include providing an extensive evaluation of different sentence embedding techniques in privacy related context which will provide valuable insights into the strengths and weaknesses of each technique. Then, the most accurate sentence embedding technique can be utilized in providing Android users with dangerous permissions rationales extracted automatically from apps' privacy policies. This would help users in making more privacy informed decisions, especially for apps that don't explain to users why they request access to users' private data.

## REFERENCES

Abdelali, A., Cowie, J., & Soliman, H. S. (2007). Improving query precision using semantic expansion. *Information Processing & Management*, *43*(3), 705–716. doi:10.1016/j.ipm.2006.06.007

Al-Kouz, A., Luca, E. W. D., & Albayrak, S. (2011). Latent semantic social graph model for expert discovery in facebook. In *11th International Conference on Innovative Internet Community Systems (I2CS 2011).* Gesellschaft für Informatik eV.

Alepis, E., & Patsakis, C. (2017, December). Hey doc, is this normal?: exploring android permissions in the post marshmallow era. In *International Conference on Security, Privacy, and Applied Cryptography Engineering* (pp. 53-73). Springer. doi:10.1007/978-3-319-71501-8_4

Altszyler, E., Ribeiro, S., Sigman, M., & Slezak, D. F. (2017). The interpretation of dream meaning: Resolving ambiguity using Latent Semantic Analysis in a small corpus of text. *Consciousness and Cognition*, *56*, 178–187. doi:10.1016/j.concog.2017.09.004 PMID:28943127

Android Developers. (2018). *Permissions overview*. Retrieved July 17, 2018, from https://developer.android.com/guide/topics/permissions/overview

Android Developers. (2019). *Location strategies*. Retrieved January 6, 2019, from https://developer.android.com/guide/topics/location/strategies

Arora, S., Liang, Y., & Ma, T. (2016). *A simple but tough-to-beat baseline for sentence embeddings*. Academic Press.

Baalous, R., & Poet, R. (2018, September). How Dangerous Permissions are Described in Android Apps' Privacy Policies? In *Proceedings of the 11th International Conference on Security of Information and Networks* (pp. 1-2). doi:10.1145/3264437.3264477

Bhatia, J., Evans, M. C., Wadkar, S., & Breaux, T. D. (2016, September). Automated extraction of regulated information types using hyponymy relations. In *2016 IEEE 24th International Requirements Engineering Conference Workshops (REW)* (pp. 19-25). IEEE. doi:10.1109/REW.2016.018

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., & Baroni, M. (2018). *What you can cram into a single vector: Probing sentence embeddings for linguistic properties*. arXiv preprint arXiv:1805.01070

Costante, E., Sun, Y., Petković, M., & den Hartog, J. (2012, October). A machine learning solution to assess privacy policy completeness: (short paper). In *Proceedings of the 2012 ACM workshop on Privacy in the electronic society* (pp. 91-96). doi:10.1145/2381966.2381979

Crocetti, G. (2015). *Textual spatial cosine similarity*. arXiv preprint arXiv:1505.03934

Federal Trade Commission. (2000). *Privacy online: Fair information practices in the electronic marketplace: A report to Congress*. Retrieved March 06, 2020, from https://www.ftc.gov/sites/default/files/documents/reports/privacy-online-fair-information-practices-electronic-marketplace-federal-trade-commission-report/privacy2000.pdf

Felt, A. P., Ha, E., Egelman, S., Haney, A., Chin, E., & Wagner, D. (2012, July). Android permissions: User attention, comprehension, and behavior. In *Proceedings of the eighth symposium on usable privacy and security* (pp. 1-14). doi:10.1145/2335356.2335360

Feng, Y., Chen, L., Zheng, A., Gao, C., & Zheng, Z. (2019). AC-Net: Assessing the Consistency of Description and Permission in Android Apps. *IEEE Access : Practical Innovations, Open Solutions*, *7*, 57829–57842. doi:10.1109/ACCESS.2019.2912210

Furnell, S., & Phippen, A. (2012). Online privacy: A matter of policy? *Computer Fraud & Security*, *2012*(8), 12–18. doi:10.1016/S1361-3723(12)70083-0

Google. (2017). *Privacy, Security, and Deception*. Retrieved December 28, 2017, from https://play.google.com/about/privacy-security-deception/personal-sensitive/

Gopinath, A. A. M., Wilson, S., & Sadeh, N. (2018). Supervised and unsupervised methods for robust separation of section titles and prose text in web documents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 850-855). doi:10.18653/v1/D18-1099

Guo, W., & Liu, H. (2013, August). The analysis of push technology based on iphone operating system. *Information and Control*, *1*, 570–574.

Handler, A. (2014). *An empirical study of semantic similarity in WordNet and Word2Vec*. Academic Press.

Harispe, S., Ranwez, S., Janaqi, S., & Montmain, J. (2015). Semantic similarity from natural language and ontology analysis. *Synthesis Lectures on Human Language Technologies*, *8*(1), 1–254. doi:10.2200/S00639ED1V01Y201504HLT027

Harkous, H., Fawaz, K., Lebret, R., Schaub, F., Shin, K. G., & Aberer, K. (2018). Polisis: Automated analysis and presentation of privacy policies using deep learning. In *27th {USENIX} Security Symposium ({USENIX} Security 18)* (pp. 531-548). USENIX.

Hosseini, M. B., Qin, X., Wang, X., & Niu, J. (2018, June). Extracting Information Types from Android Layout Code Using Sequence to Sequence Learning. *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*.

Kamaruddin, S. S., Yusof, Y., Abu Bakar, N. A., Ahmed Tayie, M., & Alkubaisi, A. A. J. (2018). Graph-based Representation for Sentence Similarity Measure: A Comparative Analysis. *International Journal of Engineering & Technology, 7*(2.14), 32-35.

Kim, S. N., Baldwin, T., & Kan, M. Y. (2010, August). Evaluating n-gram based evaluation metrics for automatic keyphrase extraction. In *Proceedings of the 23rd international conference on computational linguistics* (pp. 572-580). Association for Computational Linguistics.

Lebanoff, L., & Liu, F. (2018). *Automatic detection of vague words and sentences in privacy policies.* arXiv preprint arXiv:1808.06219

Liu, F., Fella, N. L., & Liao, K. (2016, September). Modeling language vagueness in privacy policies using deep neural networks. *2016 AAAI Fall Symposium Series*.

Lu, Y., Yu, S., Shi, M., & Li, C. (2018, August). Extract Knowledge from Web Pages in a Specific Domain. In *International Conference on Knowledge Science, Engineering and Management* (pp. 117-124). Springer. doi:10.1007/978-3-319-99365-2_10

McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (2006). A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI Magazine*, *27*(4), 12–12.

Mihalcea, R., Corley, C., & Strapparava, C. (2006, July). Corpus-based and knowledge-based measures of text semantic similarity. In Aaai (Vol. 6, No. 2006, pp. 775-780). Academic Press.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space.* arXiv preprint arXiv:1301.3781

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119). Academic Press.

Moens, M. F. (2006). *Information extraction: algorithms and prospects in a retrieval context* (Vol. 21). Springer Science & Business Media.

Olukoya, O., Mackenzie, L., & Omoronyia, I. (2020). Security-oriented view of app behaviour using textual descriptions and user-granted permission requests. *Computers & Security*, *89*, 101685. doi:10.1016/j.cose.2019.101685

Padarian, J., & Fuentes, I. (2019). Word embeddings for application in geosciences: Development, evaluation, and examples of soil-related concepts. *Soil (Göttingen)*, *5*(2), 177–187. doi:10.5194/soil-5-177-2019

Pagliardini, M., Gupta, P., & Jaggi, M. (2017). *Unsupervised learning of sentence embeddings using compositional n-gram features*. arXiv preprint arXiv:1703.02507

Pandita, R., Xiao, X., Yang, W., Enck, W., & Xie, T. (2013). {WHYPER}: Towards Automating Risk Assessment of Mobile Applications. In *Presented as part of the 22nd {USENIX} Security Symposium ({USENIX} Security 13)* (pp. 527-542). USENIX.

Perone, C. S., Silveira, R., & Paula, T. S. (2018). *Evaluation of sentence embeddings in downstream and linguistic probing tasks*. arXiv preprint arXiv:1806.06259

Qu, Z., Rastogi, V., Zhang, X., Chen, Y., Zhu, T., & Chen, Z. (2014, November). Autocog: Measuring the description-to-permission fidelity in android applications. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1354-1365). doi:10.1145/2660267.2660287

Reidenberg, J. R., Breaux, T., Cranor, L. F., French, B., Grannis, A., Graves, J. T., & Ramanath, R. et al. (2015). Disagreeable privacy policies: Mismatches between meaning and users' understanding. *Berkeley Technology Law Journal*, *30*, 39.

Schnabel, T., Labutov, I., Mimno, D., & Joachims, T. (2015, September). Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 298-307). doi:10.18653/v1/D15-1036

Slavin, R., Wang, X., Hosseini, M. B., Hester, J., Krishnan, R., Bhatia, J., & Niu, J. et al. (2016, May). Toward a framework for detecting privacy policy violations in android application code. In *Proceedings of the 38th International Conference on Software Engineering* (pp. 25-36). doi:10.1145/2884781.2884855

Steinfeld, N. (2016). "I agree to the terms and conditions":(How) do users read privacy policies online? An eye-tracking experiment. *Computers in Human Behavior*, *55*, 992–1000. doi:10.1016/j.chb.2015.09.038

Sugathadasa, K., Ayesha, B., de Silva, N., Perera, A. S., Jayawardana, V., Lakmal, D., & Perera, M. (2017, December). Synergistic union of word2vec and lexicon for domain specific semantic similarity. In *2017 IEEE International Conference on Industrial and Information Systems (ICIIS)* (pp. 1-6). IEEE. doi:10.1109/ICIINFS.2017.8300343

Sun, Y. P. (2018). *Investigating the Effectiveness of Android Privacy Policies* (Doctoral dissertation).

Tai, K. S., Socher, R., & Manning, C. D. (2015). *Improved semantic representations from tree-structured long short-term memory networks*. arXiv preprint arXiv:1503.00075

Turmo, J., Ageno, A., & Català, N. (2006). Adaptive information extraction. *ACM Computing Surveys*, *38*(2), 4. doi:10.1145/1132956.1132957

Voleti, R., Liss, J. M., & Berisha, V. (2019, May). Investigating the effects of word substitution errors on sentence embeddings. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 7315-7319). IEEE. doi:10.1109/ICASSP.2019.8683367

Wang, X., Qin, X., Hosseini, M. B., Slavin, R., Breaux, T. D., & Niu, J. (2018, May). Guileak: Tracing privacy policy claims on user input data for android applications. In *Proceedings of the 40th International Conference on Software Engineering* (pp. 37-47). doi:10.1145/3180155.3180196

Wang, X., Qin, X., Hosseini, M. B., Slavin, R., Breaux, T. D., & Niu, J. (2019). *GUILeak: Identifying privacy practices on GUI-based data*. Academic Press.

Wieting, J., Bansal, M., Gimpel, K., & Livescu, K. (2015). *Towards universal paraphrastic sentence embeddings*. arXiv preprint arXiv:1511.08198

*Rawan Baalous is a researcher at the School of Computing Science, the University of Glasgow. Her PhD focuses on the impact of privacy policy and app permission linkage on disclosure decisions. She obtained her masters degree in Information Security from the School of Computing Science, the University of Glasgow. Her main research interests include privacy policies analysis, Android permissions, usable security and privacy. Ron Poet is a lecturer in the Department of Computing Science at the University of Glasgow, specialising in Computer Security. He obtained his PhD from the Department of Maths and Theoretic Physics at the University of Cambridge, England.*