




POCASUM: policy categorizer and summarizer based on text mining and machine learning

Rushikesh Deotale³ · Shreyash Rawat³ · V. Vijayarajan³ · V. B. Surya Prasath^{1,2} 

Accepted: 26 May 2021 / Published online: 11 June 2021

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract

Having control over your data is a right and a duty that every citizen has in our digital society. It is often that users skip entire policies of applications or websites to save time and energy without realizing the potential sticky points in these policies. Due to obscure language and verbose explanations majority of users of hypermedia do not bother to read them. Further, sometimes digital media companies do not spend enough effort in stating their policies clearly which often time can also be incomplete. A summarized version of these privacy policies that can be categorized into the useful information can help the users. To solve this problem, in this work we propose to use machine learning-based models for policy categorizer that classifies the policy paragraphs under the attributes proposed like security, contact, etc. By benchmarking different machine learning-based classifier models, we show that artificial neural network model performs with higher accuracy on a challenging dataset of textual privacy policies. We thus show that machine learning can help summarize the relevant paragraphs under the various attributes so that the user can get the gist of that topic within a few lines.

Keywords Text classification · Text summarization · Privacy policy · Text mining · Machine learning · Artificial neural network

1 Introduction

Controlling and protecting your personal data is a prerogative unless you allow someone to have control over it. Privacy policies mention the regulation which the particular application promises to follow regarding the user's data. It usually mentions what kind of data they collect from you, how they use the data, under which circumstances is the data shared,

who has the access of your data, the security regarding the storage of data, the update policies and a lot more. It intends to mention beforehand of all the risk the user might undertake by agreeing to the policies of the application. The real problem lies in the inability of the user to comprehend the policy and understand the risks associated with it due to a plethora of reasons. With increasing technology, almost everyone has a smartphone today which allows even illiterate people to gain access to these applications. In general, due to their lack of technical knowledge they have to accept the policies without reading and understanding them. However, that is a relatively small percentage, the remaining majority of users of these applications and websites in spite of having the required education do simply click to agree to the policies without even reading in whole since it requires an arduous effort in understanding the verbose and technical terms utilized in these privacy policies.

The formal language of the policies along with its confusing semantics makes it irksome for the users which then accept these policies without understanding. The companies also hold the users responsible for any kind of misuse of data as the users had already agreed to the privacy policy which mentions about it. There are some applications which make it

✉ V. Vijayarajan
vijayarajan.v@vit.ac.in

✉ V. B. Surya Prasath
surya.iit@gmail.com

Rushikesh Deotale
rushikeshdeotale27@gmail.com

Shreyash Rawat
shreyash.rawat@gmail.com

¹ Division of Biomedical Informatics, Cincinnati Children's Hospital Medical Center, Cincinnati, OH 45229, USA

² Departments of Pediatrics, Biomedical Informatics, Electrical Engineering and Computer Science, University of Cincinnati College of Medicine, Cincinnati, OH, USA

³ School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India

optional for the user to go through the policy, which leads to even fewer users referring to the policy. The enormous power of data can offer insights but can also cause obstacles in terms of the huge volumes. To facilitate the users to comprehend the policies, we present a policy summarizer which shows the various attributes that have been covered by the policy and the ones which they have omitted. The attributes include the various measures or the topics that the privacy policy should include based on directives, regulations and practices that an ideal policy should incorporate. After showing the topics the policy covers, the users also get the facility to view the passage which contains the particular topic in the policy in a summarized form. Hence, this saves the user plenty of time by only reading the short summarized points present in the policy of the topics the user wishes to view. Users will more likely go through the policy if they are encountered with short summarized points related to the topics that do matter to them rather than reading huge blocks of information containing all the points in them without categorization. Reading the part of the policy which mentions about the attributes in one or two lines is much more beneficial than searching through the whole text. Users do not always go through the content of privacy policies while installing apps. The problem lies in the length of the content. To address the problems faced by the user there have been advancements in text mining that are considered to facilitate users of webpages and apps to see summarized textual data (Li et al. 2016; Izumi et al. 2007). (Costante et al. 2012) have proposed a method to resolve such issues by providing a completeness analyzer. They check the completeness and give a grade to privacy policies. This helps the users in understanding policies based on the grades given by the analyzer. They have classified each of the paragraphs present in a policy and checked whether all classes are present and give a completeness grade for the policies. Harkous et al. (2018) focused on creating a full-fledged framework so users can easily use that framework to automate the entire process. They have not only focused on low-level features but have also focused on high-level, more complex classes. Privacy policies also have options for users to opt out of some policies and take control of their data. Efforts have been made to find these particular options present in a privacy policy (Sathyen-dra et al. 2017).

Privacy policies comprise a lot of text and a lot of attributes, thus summarizing it directly would miss out salient text. Thus, to tackle that problem, classification of the text is needed based on the attributes it covers. Multiclass text classification is a classification problem in which text is classified in more than two classes (Cherfi et al. 2006). The existing solutions to the problem of multiclass classification is focused on using machine learning techniques like naive Bayes and support vector machine (SVM) (Rennie and Rifkin 2001; Silva and Ribeiro 2007). Another method is to counter the problem of multiple classification as n binary classifi-

cation and solve these problems one by one and combining the result of these binary classifications. There have been exciting progress in utilizing using neural networks and deep learning-based techniques for classifying text (Minaee et al. 2021; Satapathy et al. 2019), document modeling (Majumder et al. 2017) and for various natural language processing (NLP) tasks (Young et al. 2018). Among these techniques, convolutional and recurrent neural network-based architectures for labeling multiple categories (Chen et al. 2017), generative models for synthetically generating text for training and classifications (Li et al. 2018; Russell et al. 2019) are important. In text summarization, deep recurrent belief networks can be effectively used to model word dependencies (Chaturvedi et al. 2016). Recent improvements in deep learning involves the usage of capsule networks (Zhao et al. 2019) for challenging NLP applications such as the multilabel text classification and question answering, and long short-term (LSTM) model (Ma et al. 2018) for aspect-based sentiment analysis.

Text summarization models have been considered before the deep learning era as well. For example, Nomoto and Matsumoto (2001) present a novel approach to unsupervised text summarization. The novelty lies in exploiting the diversity of concepts in text for summarization, which has not received much attention in the summarization literature. A diversity-based approach here is a principled generalization of maximal marginal relevance (MMR) criterion by Carbonell and Goldstein (1998). They have presented a new summarization scheme where evaluation does not rely on matching extracts against human-made summaries but measuring the loss. There are many text summarization techniques presented using classical machine learning techniques like hidden Markov model (HMM), SVM and Bayes model. Another way of text summarization can be done using lexical chains (Barzilay and Elhadad 1999) formed with help of lexical cohesion. Autoencoders are also being used for text summarization (Yousefi-Azar and Hamey 2017) which used sentence ranking to extract sentences in their summary. For training the model on more specific kind of policies we can use a web-based information retrieval framework (Vijayarajan et al. 2016) to ingest peculiar policies.

In this work, we consider automatic text mining and machine learning base summarization of policies. Our policy categorizer and summarizer (POCASUM) allows us to test various plug-and-play machine learning classifiers. We test various classical machine learning models such as the K-nearest neighbors (KNN) (Abu Alfeilat et al. 2019), support vector classifier (SVC) (Suykens and Vandewalle 1999), random forests (RF) (Breiman 2001), stochastic gradient descent (SGD) (Kabir et al. 2015), along with deep learning artificial neural networks (ANNs). Experimental results on the APP-350 dataset (Zimmeck et al. 2019) indicate that

ANN driven model obtains the highest accuracy among other KNN-, SVC-, SGD- and RF-based approaches.

The rest of the work is organized as follows. Section 2 introduces our machine learning driven approach to policy categorization and summarization. Section 3 provides experimental results with different machine learning models. Section 4 concludes the paper.

2 Policy categorizer and summarization with machine learning

2.1 Aim and methodology

The goal of our proposed policy categorizer and summarizer (POCASUM) is to make it easy for users to comprehend the policies by reading short paragraphs and be clearly aware about how complete the policy is in terms of attributes that need to be mentioned in an ideal privacy policy. Hence, this includes two main tasks namely categorization and summarization. The policy paragraphs are classified under the proposed attributes so that a completeness score can be assigned to it. After checking its completeness the user might want to see the policy under a certain topic (say security). Users can then read the summarized text of the security attribute in about 2 to 3 lines which makes it tremendously easy for the user to comprehend and understand it properly.

The proposed flow of our POCASUM approach is given in Fig. 1 and consists of the following steps.

1. **Data Annotation:** The dataset has been created manually wherein each paragraph of every policy has been labeled according to the categories mentioned above.
2. **Cleaning:** Includes tokenization, stemming, lemmatization, removal of stop words.
3. **Feature Extraction:** Extracting features from text using tf-idf vectorizer or word-2-vec.
4. **Data Split:** The reduced vectors are then split into testing and training data.
5. **Dimensionality Reduction:** Number of features are reduced using LSI which uses SVD for getting the number of features to a bare minimum.
6. **Fitting Models:** Training various machine learning models to find the best potential model.
7. **Testing:** Checking the accuracy by various testing parameters.
8. **Sentence Embedding:** Creating the sentences as vector that represents the sentence semantically and syntactically.
9. **Clustering:** Clustering of the embeddings in proximity gives us the most relevant sentences.
10. **Extractive Summarization:** Picking n most relevant lines and putting it up as a paragraph.

We note that the plug-and-play methodology of our POCASUM, at the fitting models stage one can utilize various machine learning models from classical models such as KNN, SVC, RF, SGD to deep learning ANN. In what follows, we benchmark various machine learning models and test their accuracy on text summarization in privacy policies.

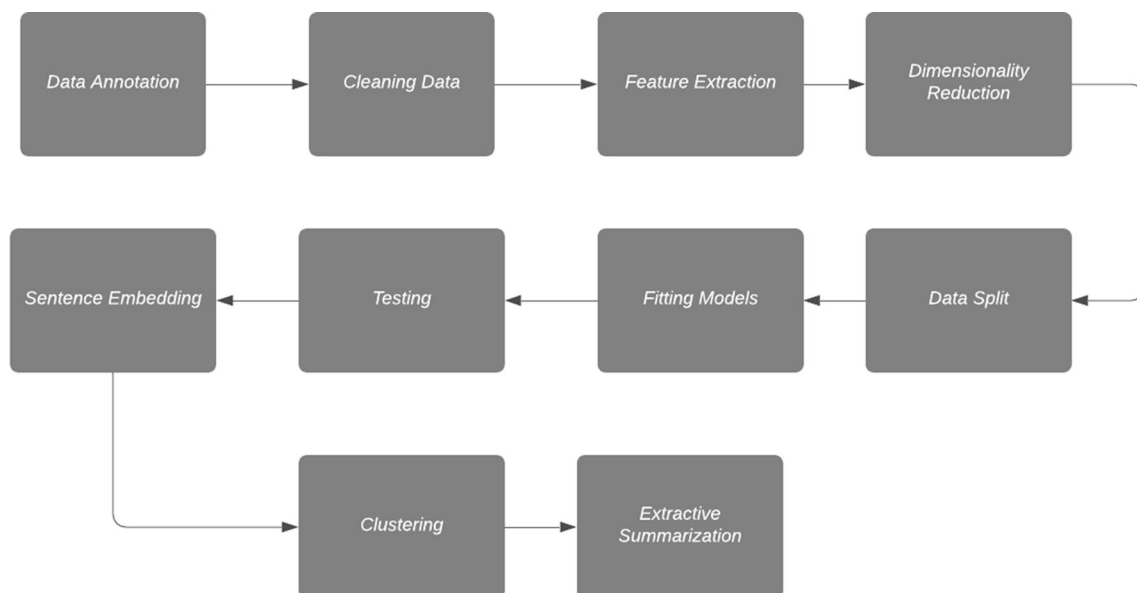


Fig. 1 Overall flow of our proposed policy categorizer and summarizer (POCASUM) approach. Here we can utilize plug-and-play machine learning classifiers at the fitting models stage

2.2 Dataset

The dataset was manually created taking into consideration 60 privacy policies of applications from the APP-350 dataset (Zimmeck et al. 2019) (<https://usableprivacy.org/data>). The policies were divided into discrete paragraphs, and each paragraph was labeled manually by the annotator. There are 5 attributes or categories that have been chosen to classify the paragraphs of the policy. The speculated important topics in a privacy policy that would affect the user's have been taken as the attributes of the analyzer. These attributes are:

- Collection: The type of personal data that the company collects from the user
- Info Usage: How is the data being utilized by the company
- Location: How is the location information being used
- Share: Under what circumstances will the users data be disclosed or shared
- Contact: Do they provide any contact information about the company

All 60 policies were labeled manually under these categories by 3 annotators in order to create the dataset. Each policy had been segmented into clusters of 2 or 3 sentences depending on the context and were then annotated within these attributes.

2.3 Text preprocessing

Text preprocessing plays a vital role in the classification of the policy statements as it provides the attributes that are to be fed into the model (Uysal and Gunal 2014). There is no perfect way of performing text preprocessing as it is subjective to the problem and the dataset available. In case of privacy policies, it was important to include compound words and headings which may not be necessary for another problem. The step-by-step process has been stated in Algorithm 1. The raw text was initially converted into tokens and the stop words were removed. The stop word removal is the process of removal of very frequently occurring words that do not contribute to the prominence of the text and was performed by a list of stop words in the English language. Then certain regular expression patterns were compiled to include certain kinds of words like email or compound words which might contribute to the prediction process.

The tokens were further normalized using a Lemmatizer and then were marshaled into a list of tokens which created our cleaned tokens. These tokens were then converted into a tf-idf vector (Fautsch and Savoy 2010) instead of Count Vectorizer as it suppresses the importance of the word occurrence and takes an egalitarian weightage of the words. This allows us to take into consideration those words that are important

Algorithm 1: Text Preprocessing

```

Input : Raw text input  $R_i$ 
Output: Vectors  $V$ 

foreach  $S \in R_i$  do
     $T = \text{Tokenize}(s)$ 
    foreach  $t \in T$  do
        if  $t$  in stop_words then
            remove  $t$ ;
        else
            continue;
        if check  $t$  with RE for compound words then
            continue;
        else
            remove  $t$ ;
         $l = \text{Lemmatize}(t)$ 
         $v = \text{tf-idf}(l)$ 
    Add  $v$  to  $V$ 

```

to the topic rather than those that appear very frequently like “an” or “the.” Figure 2 shows the flowchart of the text preprocessing steps. Due to the large number of features the vectors were further reduced by applying dimensionality reduction using latent semantic indexing (LSI) approach. Again the number of dimensions to be reduced is based on the dataset at hand, and the optimal number of dimensions can only be found through experimentation.

2.4 Modeling

After the text has been cleaned and been converted to vectors they are then taken for the model training. Constructing a multiclass text classifier involves assigning the class label to the particular vector of text that represents the features of the raw text. Supervised learning techniques were used for this purpose as the dataset had been annotated and gives it a better chance to improve the model's accuracy. We have used machine learning (Kotthoff et al. 2011) and deep learning techniques in order to get the best accuracy on multilabel classification. Under classical machine learning models, we trained four different models namely K-nearest neighbors (KNN), support vector classifier (SVC), stochastic gradient descent (SGD) and random forest (RF) classifier.

The KNN (Yang and Liu 1999; Abu Alfeilat et al. 2019) is the go-to algorithm for ubiquitous clustering problems due to its simple approach without a mathematical model and is very effective in many cases. The number of clusters is an important parameter to be set which is found out by empirical method. The elbow method is used to find out the ideal number of neighbors, but in case of supervised learning, the number of clusters are already been provided in the dataset. The SVC (Burgess 1998) is a novel technique and has proved to be immensely accurate in classification

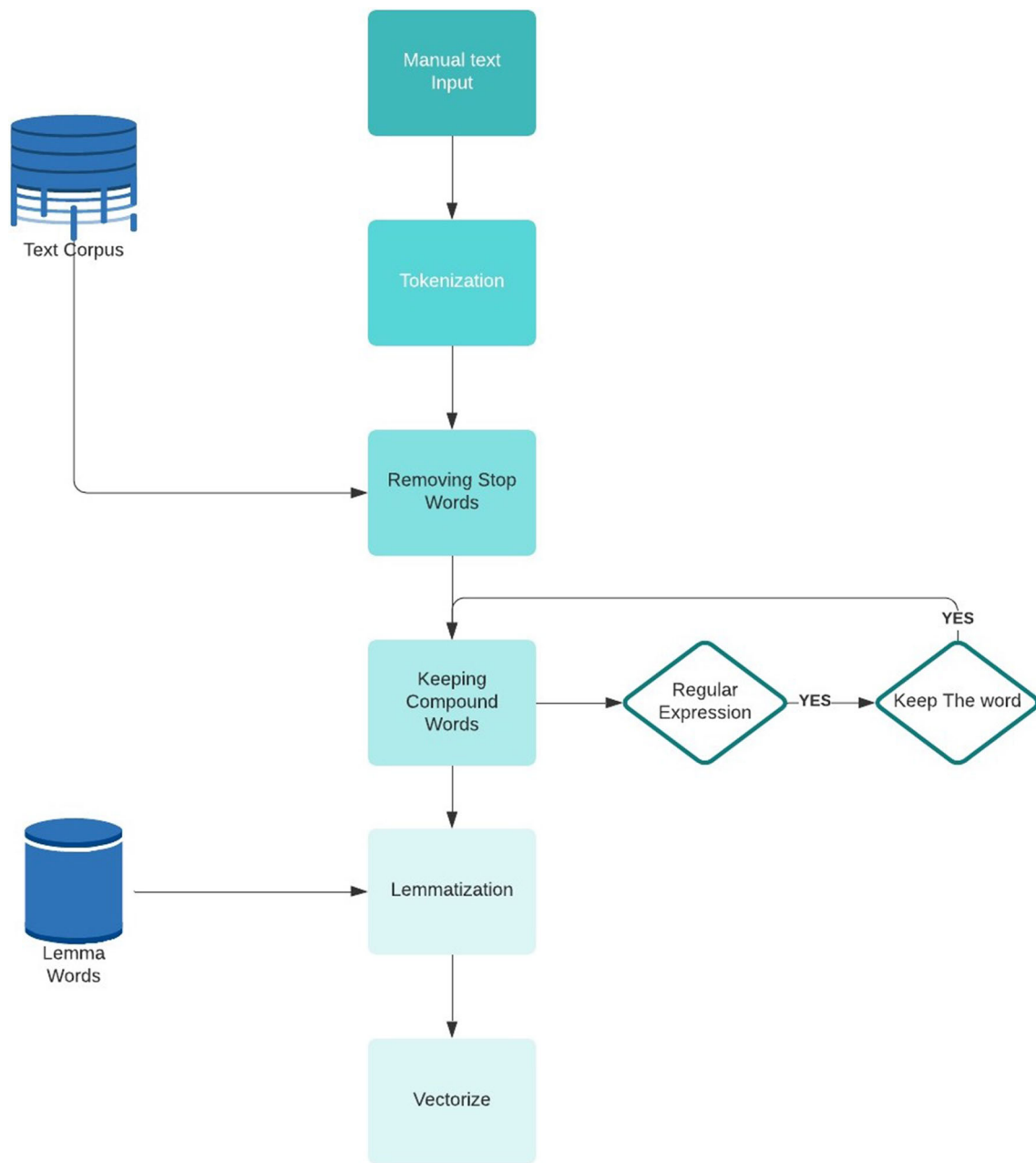


Fig. 2 Text preprocessing steps employed here for our POCASUM approach

tasks due to the less number of parameters to tune. The SGD (Kabir et al. 2015) is the most commonly used linear model and due to good reasons due to it is a good fit to the data. It aims at reducing the error by choosing the number of parameters to be changed at a given descent or step. The ensemble random forest (RF) (Breiman 2001) classifier was also used due to the multitude of decision trees giving a holistic prediction rather than a specific one leading to more chances of a higher accuracy and better prediction. We also tried out different architectures of deep artificial neural network (ANN) (Ghiassi et al. 2012)

for the multiclass classification problems due to its success in the past. The one drawback of ANN is that there are a lot of things to decide on before the model can start training. The first and foremost is the architecture of the network which is again to be decided by empirical process. After building the architecture other parameters like batch size, number of epochs, cross-validation, metrics, loss function and a lot others are to be tuned as per your requirement which is again optimally found by experimentation.

2.5 Testing

After the models have been trained on the training data, we need to test the model by giving it data that the model has never seen before and validate the results for those. There are a ton of evaluation metrics available out there. For an efficient classifier, the precision and recall are two ubiquitous evaluation metrics that one relies on. Apart from these two metrics, the F1 score and classification accuracy will also be used to judge the model. F1 score is a measure concocted by precision and recall; however, one might want to look at the specific metrics according to the type of problem.

K-fold cross-validation (Fushiki 2011) was also used for increasing the robustness of the model wherein the data are divided into k equal parts, and each part is given one iteration where k-1 parts are utilized for training and the remaining part is utilized for testing. We have used a k value as 5 for training the model. All experiments were performed in the same conditions. K-fold cross-validation improves the adaptability of the model to a great extent by increasing the variance in the test data provided to the model.

2.6 Sentence embedding

After the data have been correctly classified by the model, we need to combine the texts of all attributes in order to summarize the text of the topics. The raw combined topic wise text is to be summarized so that it can be very easy for the users to comprehend the meaning of it. Just as we represent each word as a vector in a tf-idf or any other vectorizer, in order to carry out extractive summarization each sentence needs to be represented as a vector (Ahamad 2019). That vector should be made by taking the syntactic, semantic and all the important properties of the sentence it represents. Sentence embedding are those vectors that represent the sentence taking its various properties so that we can get to know the proximity between sentences. Hence, sentences that share similar semantic and syntactic properties lead to similar vector representation (Barzilay and Elhadad 1999).

The sentence embedding was facilitated using skip-thought vectors which use an encoder–decoder model for encoding sentences into vectors. The encode maps the words to a sentence vector and the decoder generates the sentences surrounding the current sentence. ConvNet and RNN were specifically used to create encoder and decoder models which generate sentence embeddings. Now these sentences can also

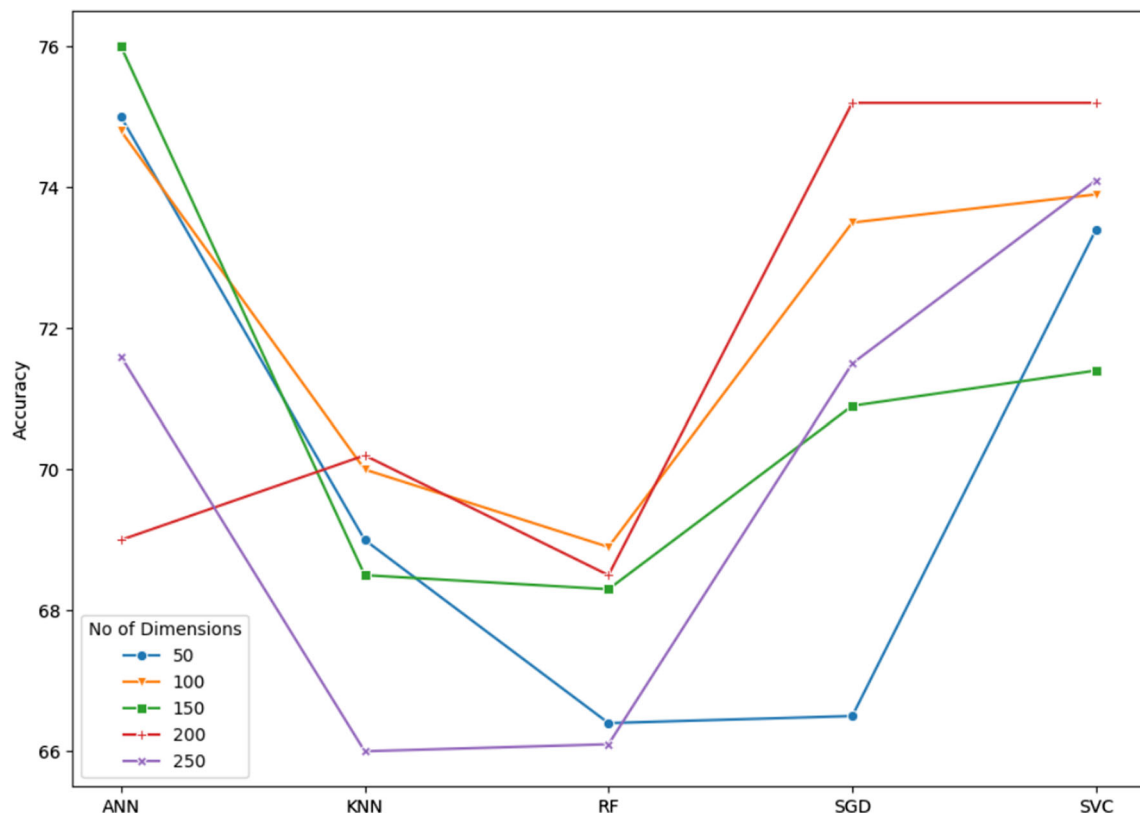


Fig. 3 Accuracy of different machine learning classifiers with respect to the number of dimensions. Here, we tested K-nearest neighbors (KNN), support vector classifier (SVC), stochastic gradient descent (SGD) and random forest (RF) and artificial neural network (ANN) classifiers

be represented on a graph as a vector which will enable us to see the characteristics or the surroundings of the vector. Since we only require the sentence embeddings and not the surrounding sentences we will only utilize the encoder part of the model as explained using Algorithm 2.

Algorithm 2: Sentence Embedding

Input : Words $\{w_i : W \in s\}$, word embedding $(x_i)^t$ denote its word embeddings, a set of sentences S

Output: Sentence Embeddings $\{v_s : s \in S\}$

foreach $s \in S$ **do**

foreach $W \in s$ **do**

$$r^t = \sigma(W_r x^t + U_r h^{t-1})$$

$$z^t = \sigma(W_z x^t + U_z h^{t-1})$$

$$\bar{h}^t = \tanh(W_x x^t + U(r^t \odot h^{t-1}))$$

$$h^t = (1 - z) \odot h^{t-1} + z^t \odot \bar{h}^t$$

The hidden state $(h_i)^t$ thus represents the full encoded sentence embedding

2.7 Clustering

Once the sentences have been converted into vectors, now we need to take n most relevant sentences from the sample space by using an unsupervised clustering algorithm (Bennani-Smires et al. 2018; Valdivia et al. 2020). These embeddings are clustered in high-dimensional vector space into a pre-defined number of clusters according to the number of sentences required in the summary of the text. K-nearest neighbors (KNN) (Abu Alfeilat et al. 2019) was used to cluster the vectors and the top 3 vectors were chosen. The measure used to cluster was Euclidean distance; however, many other metrics are also present. These vectors were then mapped to their text and their corresponding sentences are extracted and displayed as a paragraph. This gives us the extractive summary of the particular attribute the user wishes to read.

3 Experimental results

We tested different machine learning classifiers in our POCA-SUM approach, namely, the K-nearest neighbors classifier (KNN), support vector classifier (SVC), stochastic gradient descent (SGD), random forest (RF) and artificial neural network (ANN) models. There have been a plethora of empirical methods that have been used throughout the process. Starting from what kind of words to keep, trying out various cleaning techniques, number of dimensions and the features that must be reduced to all experiments, have been tried out to get the best possible results. Different dimensions were for each classification model as shown in Fig. 3. Even the kind of

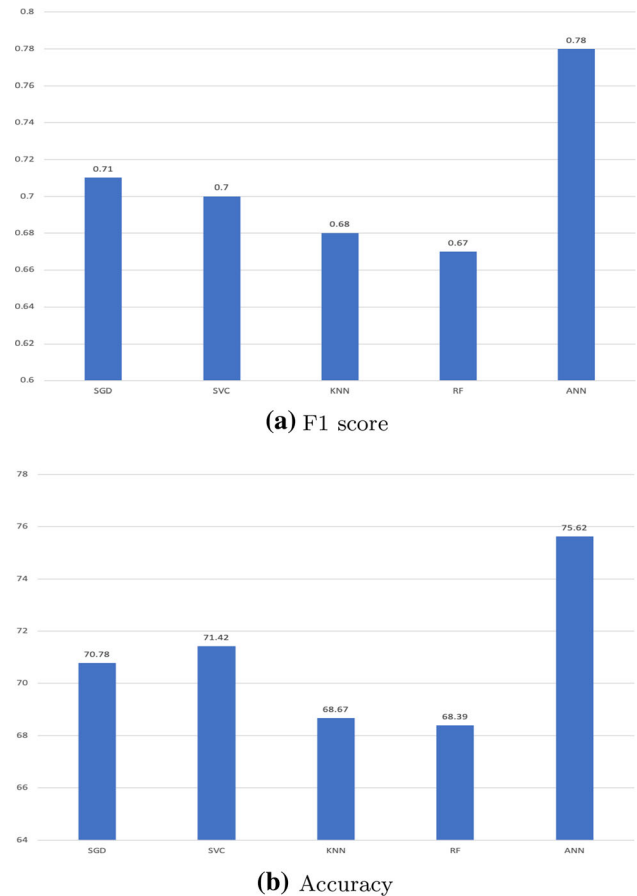


Fig. 4 Comparison of different classical (KNN, SVC, RF, SGD) and deep ANN machine learning models in terms of **a** F1 score and **b** accuracy

machine learning model to choose, the parameters on which the model is being tuned to and the architecture of the neural network are all subjective things that do not provide ideal answers for the input data. These choices vary from problem to problem and carrying out empirical methods is the only way to get an idealized value. As can be seen, the performance of KNN, RF and SGD are lower, in general than the SVC and ANN.

In terms of the performance of various machine learning classifiers, our experimental testing was done with classical—KNN, SVC, RF, SGD, as well as various architectures of ANN. Among the obtained F1 scores of KNN (68%), SVC (70%), SGD (71%) and RF (67%), SGD proved out to be the best machine learning model in our sample space with an F1 score of 71%. The deep learning ANN, however, performed very well with an F1 score of 78% and an accuracy score of 75.62, thus outperforming the rest of the models. Figure 4 shows the performance of various models with F1 score and accuracy. To assess the summarizer, we took a random policy on which the model had not been trained on. The policy was then segmented into chunks of 2 to 3 lines and

| | |
|--|---|
| <p>' A. WHAT KIND OF INFORMATION WE COLLECT a. Personal Information. We do not collect Personal Information. "Personal Information" is information that identifies you or another person, which may be transmitted or received when you use an Application, the Services and/or the Site. Personal Information includes your names, physical addresses, email addresses, telephone, fax, SSN, information stored within your Device and other information you transmit or receive using an Application, the Service and or the Site which identifies you or another person.b. Non-personal identification. We may collect non-personal identification information about installed applications, application usage information and device information. The information you give us, for example, when you give us your opinions to our application and services via our feedback channel, such as your email address, and names;'</p> <p>'[1] Information We Collect and Use: The Personal Information you provide is used for such purposes as answering questions, improving the content of the Services, customizing the advertising and content you see, and communicating with you about Company's products and services, including specials and new features. Personal Information You Provide to Us: We receive and store any information you enter on our Services or provide to us in any other way, including via any third party site or service through which you authorize us to access your information (e.g., Facebook). The types of Personal Information collected in this fashion may include but not limited to, an individual's name, phone number, credit card or other billing information, email address and home address. This Privacy Policy in no way limits or restricts our collection of aggregate or anonymous information. In this Privacy Policy, we refer to all information collected from or about you, including personal information and non-personal information, as "Your Information." Use of Applications: When you launch any of our applications, we collect information regarding your device type, operating system and version, carrier provider, IP address, Media Access Control (MAC) address, International Equipment Mobile ID (IMEI), whether you are using a point package, the game version, the device's geo-location, language settings, and unique device ID. In addition, we create a unique user ID to track your use of our Service. This unique user ID is stored in connection with your Account profile information to enable us to move Your Information to a new device at your request. In addition, it may be used to link a character with which you play our games to your username on the Forums. When you play our games, we also collect information about your play and interaction with other users and the Service.'</p> <p>'Information You Give Us We may collect information you choose to provide to us, such as your name, email address, mobile phone number, your photo, your friends' contact information as stored on your phone, and the content that you create, such as your texts, photos, and videos. If, as part of your use of the Service, you connect your Service account with an account from a supported social network account, such as Facebook (a "Social Media Account"), we may receive personal information from the corresponding social network that relates to your Social Media Account. Such personal information may be about you and/or your Social Media Account contacts. Please check the policies of the applicable social network in order to understand what information we receive. Even if you later disconnect your Service account from your Social Media Account, we still keep a copy of the personal information (such as your Social Media Account profile photo and your contact list) that we received from the connection of the two accounts, so that we may more easily connect you with your Social Media Account contacts who register with the Service in the future. Information We Get from Others We may get information about you from other sources, including from social networks you link to through the Service. We may add this to information we get from the Service. Information Automatically Collected We use a third-party Service Provider (defined below) to automatically log information about you and your computer or mobile device, and how you use and interact with the Service. For example, when you access the Service, we may log your operating system type, browser type and language, the pages you viewed, how long you spent on a page, access times, Internet protocol (IP) address, your mobile device ID, mobile device serial number, unique user ID, wireless carrier, and information about your use of and actions on the Service. We may also use third party advertisements to support our Service. Some of these advertisers may use technology such as cookies, web beacons, pixel tags, or log files when they advertise on our Service, which may send these advertisers information, including your non-personal information.'</p> | <p>'Personal Information. We do not collect Personal Information. A. WHAT KIND OF INFORMATION WE COLLECT a. We may collect non-personal identification information about installed applications, application usage information and device information. The information you give us, for example, when you give us your opinions to our application and services via our feedback channel, such as your email address, and names; Non-personal identification.'</p> <p>'In this Privacy Policy, we refer to all information collected from or about you, including personal information and non-personal information, as "Your Information." When you play our games, we also collect information about your play and interaction with other users and the Service. The types of Personal Information collected in this fashion may include but not limited to, an individual's name, phone number, credit card or other billing information, email address and home address. This Privacy Policy in no way limits or restricts our collection of aggregate or anonymous information. In addition, it may be used to link a character with which you play our games to your username on the Forums.'</p> <p>'Such personal information may be about you and/or your Social Media Account contacts. Please check the policies of the applicable social network in order to understand what information we receive. If, as part of your use of the Service, you connect your Service account with an account from a supported social network account, such as Facebook (a "Social Media Account"), we may receive personal information from the corresponding social network that relates to your Social Media Account. Some of these advertisers may use technology such as cookies, web beacons, pixel tags, or log files when they advertise on our Service, which may send these advertisers information, including your non-personal information. We may also use third party advertisements to support our Service.'</p> |
| (a) Input | (b) Output |

Fig. 5 Examples of POCASUM with ANN result from the APP-350 data set. **a** Input and **b** output from the system

were fed into the ANN model for classification as shown in the input text Fig. 5. For testing purposes, we took up the first category which was the type of information the particular company collected. All the sentences from the policy that belong to this class were combined and appended. Then these sentences were then fed into the summarizer to check whether the results semantically made sense. The results shown in Fig. 5 turned out to be comprehensible which covered the main points of the text paragraph, hence making it convenient for the user.

We have tried and tested different architectures for the ANN model for the classification task, and ANN with architecture (64+32+6) as in Table 1 proves out to be the superior of them all which obtained 78% F1 score. The performance of this best ANN model in terms of training and validation is shown in Fig. 6. Out of all available machine learning models, SGD turned out to be the accurate classifier among classical

machine learning models as shown in Table 2. As can be seen, the SGD model obtained reduced F1 score as the number of classes increased. Overall, the ANN model within POCASUM obtained the best performance indicating promise in obtaining comprehensible text summarization. However, our tested ANN architecture as shown in Table 1 is limited to 3 or 4 layers and unlike typical “deeper” architectures considered in other works in text mining and visual computing. The reason for this is the limitation of the dataset we considered here, deeper layers tend to overfit and designing an optimal ANN remains a challenge.

In summary, in this work we have provided a framework to analyze the authenticity of a privacy policy by carrying out multiclass classification and summarization techniques. Further, we have combined the text that belongs to a particular class or attribute so that the paragraphs can be summarized and only the necessary information is presented to the user

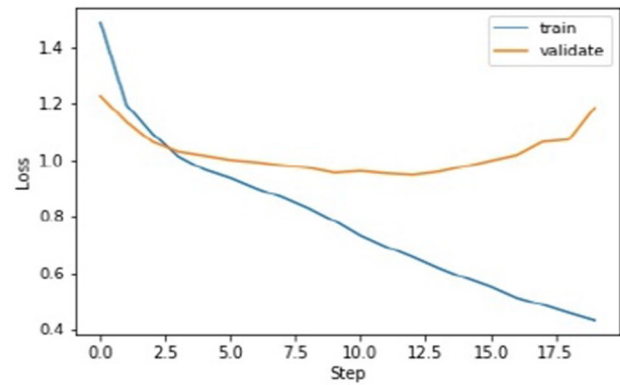
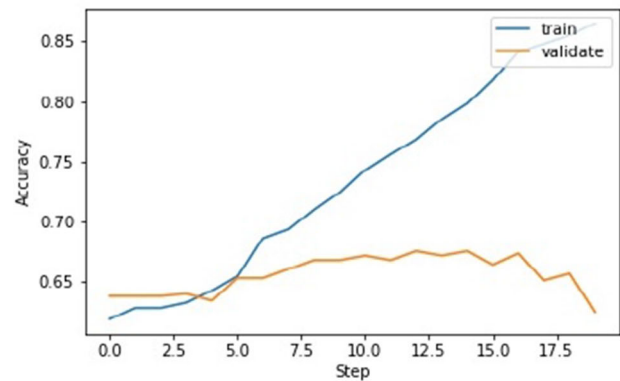
Table 1 Different architectures of ANN tested in our POCASUM approach for text summarization

| Architecture | F1 score | Precision (%) |
|-------------------|----------|---------------|
| 64 + 32 + 6 | 0.78 | 76 |
| 64 + 32 + 32 + 6 | 0.73 | 72.4 |
| 32 + 32 + 6 | 0.72 | 72.35 |
| 128 + 64 + 32 + 6 | 0.69 | 70.1 |
| 128 + 32 + 6 | 0.70 | 70.9 |

which makes it easy for them to comprehend the policy. Despite the positive results obtained with ANN combined POCASUM, there is still improvements that can be done to make this work more convenient for users: (i) the tested architectures represent our experimental heuristics and model reduction within the ANN context remains to be tested further and (ii) an expanded dataset that includes other textual policies are required to study the feasibility of machine learning driven POCASUM approach. Moreover, semantic analysis of sentences (Tur et al. 2012; Nguyen et al. 2019) is one area which we might be lacking in NLP-based text summarization considered here. Due to the changing dynamics of the language and several ways to write the same sentence, it is very difficult to make the algorithm understand the meaning behind the sentence. If semantic analysis is facilitated within our POCASUM approach, it will make it very convenient for users as we can directly tell them whether a certain topic abides by the rules mentioned in an ideal privacy policy. We can directly assign a score which will tell them how safe the policy is, and the users will not even have to see the content of the policy and will get to know how safe the policy is automatically.

4 Conclusion

In this work, we used text mining and machine learning models for policy notices categorization and summarization to aid users with relevant information in a succinct way. Our policy categorizer and summarizer (POCASUM) approach can aid the users of hypermedia to be more careful about their rights and data and makes it easy for the users to understand the digital media company's motives. We benchmarked different plug-and-play machine learning classifiers including KNN, SVM, RF, SGD and deep ANN models. Our experimental on a dataset of policies indicate that ANN-based model performs well with good accuracy. Our initial experiments indicate that we can reduce the time taken to analyze a policy from 15 to 20 min down per user to a few seconds. This indicates that deeper networks can be used in our POCASUM approach, thereby improving the summarization capabilities; however, this requires a bigger dataset for training the learn-

**(a) Loss****(b) Accuracy****Fig. 6** Performance of the best ANN model (64 + 32 + 6 architecture) with epoch steps in terms of training and validation **a** loss and **b** accuracy**Table 2** Classification with stochastic gradient descent (SGD) for different classes

| Classes | F1 score | Precision | Recall | Support |
|---------|----------|-----------|--------|---------|
| 0 | 0.79 | 0.78 | 0.814 | 340 |
| 1 | 0.44 | 0.41 | 0.483 | 60 |
| 2 | 0.39 | 0.41 | 0.37 | 58 |
| 3 | 0 | 0 | 0 | 4 |
| 4 | 0.52 | 0.58 | 0.47 | 44 |
| 5 | 0.52 | 0.65 | 0.43 | 30 |

ing model. Further, benchmarking this reduction of reading time requires a deeper user-based feedback study. Another area of improvement can be where sentence embedding is executed. Apart from skip thought vectors there are many other embedding techniques that can be used to better represent the sentences as vectors like skip gram vectors and quick thought vectors. Quick thought vectors (Russell et al. 2019) is a recent development in skip thought vectors in which the task of forecasting the next sentence given the previous one is handled as a classification problem.

Funding VBSP is supported by NCATS/NIH Grant U2CTR002818, NHLBI/NIH Grant U24HL148865, NIAID/NIH Grant U01AI150748, Cincinnati Children's Hospital Medical Center–Advanced Research Council (ARC) Grants 2018–2020 and the Cincinnati Children's Research Foundation–Center for Pediatric Genomics (CPG) Grants 2019–2021.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

- Abu Alfeilat HA, Hassanat AB, Lasassmeh O, Tarawneh AS, Alhasanat MB, Eyal Salman HS, Prasath VS (2019) Effects of distance measure choice on K-nearest neighbor classifier performance: a review. *Big Data* 7(4):221–248
- Ahamad A (2019) Generating text through adversarial training using skip-thought vectors. In: Annual conference of the north American chapter of the association for computational linguistics, pp 53–60
- Barzilay R, Elhadad M (1999) Using lexical chains for text summarization. *Advances in Automatic Text Summarization*, pp 111–121
- Bennani-Smires K, Musat C, Hossmann A, Baeriswyl M, Jaggi M (2018) Simple unsupervised keyphrase extraction using sentence embeddings. In: 22nd conference on computational natural language learning (conll), pp 221–229
- Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32
- Burges CJ (1998) A tutorial on support vector machines for pattern recognition. *Data Min Knowl Disc* 2(2):121–167
- Carbonell J, Goldstein J (1998) The use of Mmr, diversity-based reranking for reordering documents and producing summaries. In: 21st annual international Acm Sigir conference on research and development in information retrieval, pp 335–336
- Chaturvedi I, Ong Y-S, Tsang IW, Welsch RE, Cambria E (2016) Learning word dependencies in text by means of a deep recurrent belief network. *Knowl Based Syst* 108:144–154
- Chen G, Ye D, Xing Z, Chen J, Cambria E (2017) Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. In: 2017 international joint conference on neural networks (IJCNN), pp 2377–2383
- Cherfi H, Napoli A, Toussaint Y (2006) Towards a text mining methodology using association rule extraction. *Soft Comput* 10(5):431–441
- Costante E, Sun Y, Petković M, den Hartog J (2012) A machine learning solution to assess privacy policy completeness. In: ACM workshop on privacy in the electronic society, pp 91–96
- Fautsch C, Savoy J (2010) Adapting the tf idf vector-space model to domain specific information retrieval. In: ACM symposium on applied computing, pp 1708–1712
- Fushiki T (2011) Estimation of prediction error by using k-fold cross-validation. *Stat Comput* 21(2):137–146
- Ghiassi M, Olschimke M, Moon B, Arnaudo P (2012) Automated text classification using a dynamic artificial neural network model. *Expert Syst Appl* 39(12):10967–10976
- Harkous H, Fawaz K, Lebrete R, Schaub F, Shin KG, Aberer K (2018) Polisis: automated analysis and presentation of privacy policies using deep learning. In: 27th usenix security symposium, pp 531–548
- Izumi K, Matsui H, Matsuo Y (2007) Integration of artificial market simulation and text mining for market analysis. *Soft Computing* 1199–1205
- Kabir F, Siddique S, Kotwal MRA, Huda MN (2015) Bangla text document categorization using stochastic gradient descent (SGD) classifier. In: International conference on cognitive computing and information processing, pp 1–4
- Kotthoff L, Gent IP, Miguel I (2011) A preliminary evaluation of machine learning in algorithm selection for search problems. In: Fourth annual symposium on combinatorial search, Barcelona, Catalonia, Spain
- Li J, Fong S, Zhuang Y, Khoury R (2016) Hierarchical classification in text mining for sentiment analysis of online news. *Soft Comput* 20(9):3411–3420
- Li Y, Pan Q, Wang S, Yang T, Cambria E (2018) A generative model for category text generation. *Inf Sci* 450:301–315
- Ma Y, Peng H, Khan T, Cambria E, Hussain A (2018) Sentic lstm: a hybrid network for targeted aspect-based sentiment analysis. *Cogn Comput* 10(4):639–650
- Majumder N, Poria S, Gelbukh A, Cambria E (2017) Deep learning-based document modeling for personality detection from text. *IEEE Intell Syst* 32(2):74–79
- Minaee S, Kalchbrenner N, Cambria E, Nikzad N, Chenaghlu M, Gao J (2021) Deep learning based text classification: a comprehensive review
- Nguyen HT, Duong PH, Cambria E (2019) Learning short-text semantic similarity with word embeddings and external knowledge sources. *Knowl Based Syst* 182:104842
- Nomoto T, Matsumoto Y (2001) A new approach to unsupervised text summarization. In: 24th annual international ACM SIGIR conference on research and development in information retrieval, pp 26–34
- Rennie JD, Rifkin R (2001) Improving multiclass text classification with the support vector machine, Technical report no. 210. MIT Artificial Intelligence laboratory, Cambridge, MA, USA
- Russell D, Li L, Tian F (2019) Generating text using generative adversarial networks and quick-thought vectors. In: IEEE international conference on computer and communication engineering technology, pp 129–133
- Satapathy R, Li Y, Cavallari S, Cambria E (2019) Seq2seq deep learning models for microtext normalization. In: 2019 international joint conference on neural networks (IJCNN), pp 1–8
- Sathyendra KM, Wilson S, Schaub F, Zimmeck S, Sadeh N (2017) Identifying the provision of choices in privacy policy text. In: 2017 conference on empirical methods in natural language processing, pp 2774–2779
- Silva C, Ribeiro B (2007) On text-based mining with active learning and background knowledge using svm. *Soft Comput* 11(6):519–530
- Suykens JA, Vandewalle J (1999) Least squares support vector machine classifiers. *Neural Process Lett* 9(3):293–300
- Tur G, Deng L, Hakkani-Tür D, He X (2012) Towards deeper understanding: Deep convex networks for semantic utterance classification. In: IEEE international conference on acoustics, speech and signal processing, pp 5045–5048
- Uysal AK, Gunal S (2014) The impact of preprocessing on text classification. *Inf Process Manag* 50(1):104–112
- Valdivia A, Martinez-Camara E, Chaturvedi I, Luzón MV, Cambria E, Ong Y-S, Herrera F (2020) What do people think about this monument? Understanding negative reviews via deep learning, clustering and descriptive rules. *J Ambient Intell Hum Comput* 11(1):39–52
- Vijayarajan V, Dinakaran M, Tejaswin P, Lohani M (2016) A generic framework for ontology-based information retrieval and image retrieval in web data. *Hum Cent Comput Inf Sci* 6(1):18

- Yang Y, Liu X (1999) A re-examination of text categorization methods. In: 22nd annual international ACM SIGIR conference on research and development in information retrieval, pp 42–49
- Young T, Hazarika D, Poria S, Cambria E (2018) Recent trends in deep learning based natural language processing. *IEEE Comput Intell Mag* 13(3):55–75
- Yousefi-Azar M, Hamey L (2017) Text summarization using unsupervised deep learning. *Expert Syst Appl* 68:93–105
- Zhao W, Peng H, Eger S, Cambria E, Yang M (2019) Towards scalable and reliable capsule networks for challenging NLP applications. In: 57th annual meeting of the association for computational linguistics, pp 1549–1559
- Zimmeck S, Story P, Smullen D, Ravichander A, Wang Z, Reidenberg J, Russell N, Cameron, Sadeh N (2019) MAPS: scaling privacy compliance analysis to a million apps. *Proc Priv Enhanc Technol* 2019(3):66–86

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.