Voice Privacy Assistant for Monitoring In-home Voice Commands

Bang Tran[1], Xiaohui Liang[1], Gabriel Ghinita[2], Caroline Summerour[3], and John A. Batsis[3]
[1]*University of Massachusetts Boston, MA, USA*
[2]*Hamad Bin Khalifa University, Doha, Qatar*
[3]*University of North Carolina at Chapel Hill, NC, USA*

Abstract

Voice assistant systems (VAS), such as Google Assistant or Amazon Alexa, provide convenient means for users to interact verbally with online services and control smart home devices. Voice commands contain highly-sensitive information about individuals, and sharing such data with service providers must be done in a carefully controlled and transparent manner in order to prevent privacy breaches. We introduce a framework named VPASS that supports the management of *personalized* privacy requirements for VAS. Our mechanisms employ deep transfer learning techniques for processing voice commands and can accurately detect privacy-sensitive commands based on an individual's prior history of VAS interaction. VPASS continuously analyzes the privacy risks and generates monthly reports or immediate alerts based on user-defined policies.

1 Introduction

Voice Assistant Systems (VAS), such as Google Assistant and Amazon Alexa, gained huge popularity in the past decade [5]. Despite their benefits, VAS devices raise significant privacy concerns. As VAS continues to gain popularity and become an integral part of various aspects of daily life, there is an increased potential for malicious actors to gain access to voice data, compromising user privacy [4]. These VAS devices rely on Voice Service Providers (VSP) to process and interpret user commands, which requires sending user data to remote servers for analysis [3]. In this work, we investigate privacy disclosure based on the voice commands' contents and context. We propose a framework VPASS that takes into account two perspectives of privacy leakage, namely information disclosure and privacy sensitivity. VPASS measures the information disclosure by checking whether the semantic information of the current command has been disclosed to the VSP in previous use. VPASS employs a customized deep transfer learning [9] model to infer the privacy sensitivity of each command. VPASS notifies users using either monthly reports or immediate alerts, according to the user-defined policies on the results of information disclosure and privacy sensitivity.

2 VPASS framework

VPASS can be a smartphone app set up by a user using the same account and password that were used to register the VAS device to the VSP. VPASS continuously downloads the voice command transcripts from the VSP and monitors the privacy risks of the voice commands. As shown in Figure 1, VPASS has three components, information disclosure analysis, privacy sensitivity analysis, and user notification.

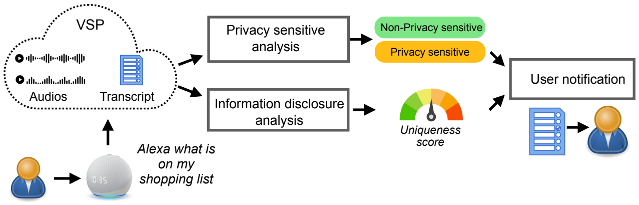
VSP
Audios
Transcript
Privacy sensitive analysis
Non-Privacy sensitive
Privacy sensitive
Information disclosure analysis
Uniqueness score
User notification
Alexa what is on my shopping list

Figure 1: VPASS overview

2.1 Information disclosure analysis

To determine how much new information a voice command discloses to the VSP, VPASS calculates the uniqueness of a given command compared to other commands previously issued by the user. VPASS incorporates a semantic similarity measure using BERT [6]. Specifically, two BERT embedding vectors v_i, v_j of two voice commands $c_i, c_j (1 \le i, j \le n+1)$ are used to calculate the cosine-similarity score [8] $s_{i,j} = \mathsf{Sim}(v_i, v_j) = \frac{v_i \cdot v_j}{\|v_i\| \cdot \|v_j\|}$. VPASS then calculates a uniqueness score of command c_{n+1} using the most similar command from all previous n commands $\mathcal{C}_n$: $\mathsf{Uni}(c_{n+1}, \mathcal{C}_n) = 1 - \max_{c_j \in \mathcal{C}_n} s_{n+1,j}$. VPASS provides another option for calculating uniqueness score for time-sensitive commands; it derives a subset of commands $\mathcal{C}_k \subseteq \mathcal{C}_n$ that includes recent k-day commands and calculates $\mathsf{Uni}(c_{n+1}, \mathcal{C}_k) = 1 - \max_{c_j \in \mathcal{C}_k} s_{n+1,j}$.

2.2 Privacy sensitivity analysis

To determine whether voice commands contain sensitive information, we need to address two challenges: the sensitivity of a voice command should be determined at both word-level

This work is funded by R01-AG067416 and IIS-1909806.

and context-level, and sensitivity is difficult to evaluate using common rules. We thus explore deep learning models to evaluate the privacy sensitivity of voice commands.

We conducted an IRB-approved study to collect 14 months (from January 2022 to February 2023) of in-home Alexa usage data from 15 older adults [1]. Each older adult received \$20 per session and was encouraged to use the device daily with an incentive of \$5 per month. We collected a total number of in-home voice commands (n=28,598). We employed five human annotators to label commands with sensitive or non-sensitive labels, and each was paid \$100. We designed a guideline to determine sensitive commands based on sensitive topics [10], and then provided the guideline to the annotators We realized the labeling effort is huge and needed them to label overlapped commands to ensure reliability. Thus, we removed the similar commands using a BERT-based similarity threshold ($s_{i,j} \geq 0.88$) to reduce the number of commands to 3,667. These commands were labeled by each annotator. Using majority voting, we obtained 794 sensitive and 2,873 non-sensitive commands. Finally, based on similarity, we extend labels to the whole dataset (7,376 sensitive and 21,222 non-sensitive). To balance the sensitive/non-sensitive classes, we synthesized sensitive commands [7] using the SpaCy Toolkit [2] to identify the keywords and using BERT to replace the keywords with new ones while maintaining the commands' integrity and ensuring the new commands highly similar to the original command.

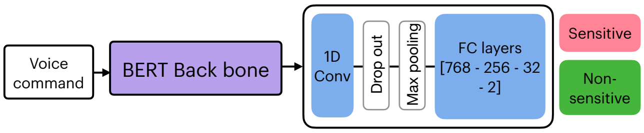

Figure 2: Sensitivity inference model

We propose a sensitivity inference model that uses the BERT embedding of a voice command as input and outputs sensitive labels, as shown in Figure 2. The BERT backbone is connected to the 1D convolution layer, a dropout layer, a pooling layer, a flattening layer, and four fully-connected layers. The final outputs are the labels, either sensitive or non-sensitive. We trained this model with 80% of manually labeled commands, extended commands, and synthesized commands, and tested it with 20% of manually labeled commands. The training and testing have been conducted for five independent rounds, and the testing accuracy on the 20% manually labeled commands is 93.87% and the balanced dataset with the synthesized commands help achieve balanced results in F1-score, precision and recall, as shown in Table 1.

2.3 User notification

VPASS notifies users of monthly reports or immediate alerts. In the monthly report, the unique scores of the commands in the past month are presented such that users can easily identify the commands that are significantly more unique than others. An example of *monthly report* from User 008 is shown in Figure 3 where the uniqueness scores of 7 commands out of 100 in May 2022 are larger than 0.4. We further incorporated sensitivity inference results into the monthly report. If a command is sensitive, its bar color is red; otherwise, its bar color is cyan. Uniqueness score 0 is replaced with -0.1 to ensure a bar for these commands. In this example, four sensitive commands were inferred using our model. These commands are highlighted in the figure for users to review easily. For the *privacy alert*, users can define the alert policy according to the uniqueness and sensitive inference: i) if a command has uniqueness score $\geq th$; ii) if a command is determined as sensitive; or iii) if a command has uniqueness score $\geq th$ and is sensitive. In this example, the first policy triggers 7 alerts, the second policy triggers 4 alerts, and the third policy triggers 2 alerts. VPASS allows users to customize the policies to balance privacy risks and management efforts.

Table 1: Five rounds of testing results

	Accuracy	F1-score		Precision		Recall	
		class 0	class 1	class 0	class 1	class 0	class 1
R_1	0.94	0.951	0.915	0.966	0.891	0.937	0.941
R_2	0.95	0.965	0.888	0.971	0.871	0.960	0.905
R_3	0.91	0.931	0.830	0.948	0.811	0.931	0.851
R_4	0.95	0.969	0.875	0.960	0.909	0.978	0.843
R_5	0.95	0.967	0.881	0.974	0.858	0.595	0.907
Ave.	**0.9387**	**0.958**	**0.878**	**0.964**	**0.868**	**0.953**	**0.889**

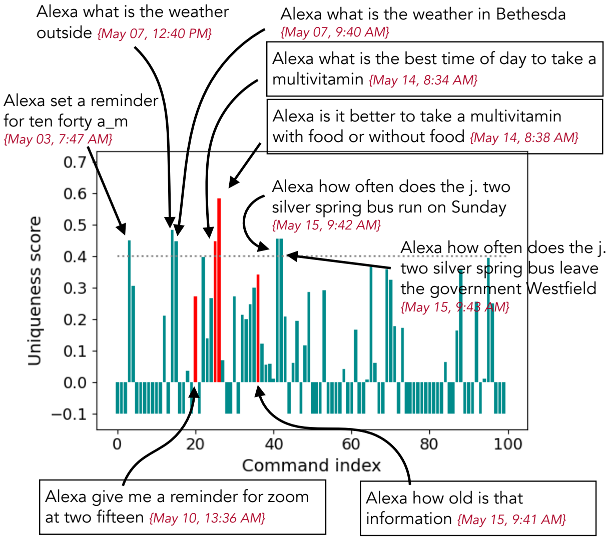

Figure 3: Monthly report of user 008 in May 2022: 30-day history, $th = 0.4$. Boxed commands are sensitive commands.

3 Conclusion

We introduced a framework VPASS for users to manage personalized privacy requirements for VAS. VPASS evaluates each voice command's information disclosure and privacy sensitivity, and notifies users with monthly reports or immediate alerts to present critical information with an intuitive interface. Through real-data evaluation, VPASS has shown high accuracy of the uniqueness and sensitivity inference.

References

[1] Project: Exploiting voice assistant systems for early detection of cognitive decline. https://cogvox.org [accessed 25-May-2023].

[2] Spacy 101: Everything you need to know. https://spacy.io/usage/spacy-101 [accessed 27-April-2023].

[3] Deeksha Anniappa and Yoohwan Kim. Security and privacy issues with virtual private voice assistants. In *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0702–0708. IEEE, 2021.

[4] Daniel Bermuth, Alexander Poeppel, and Wolfgang Reif. Jaco: An offline running privacy-aware voice assistant. In *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 618–622. IEEE, 2022.

[5] Peng Cheng and Utz Roedig. Personal voice assistant security and privacy—a survey. *Proceedings of the IEEE*, 110(4):476–507, 2022.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Alberto Fernández, Salvador Garcia, Francisco Herrera, and Nitesh V Chawla. Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, 61:863–905, 2018.

[8] Pang-Ning Tan Michael Steinbach Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005. ISBN 0-321-32136-7, chapter 8; page 500.

[9] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27*, pages 270–279. Springer, 2018.

[10] Rahul Tripathi, Balaji Dhamodharaswamy, Srinivasan Jagannathan, and Abhishek Nandi. Detecting sensitive content in spoken language. In *2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 374–381. IEEE, 2019.

A Similarity and uniqueness scores

To reduce the labeling effort, we asked annotators to label non-semantic-similar commands only. In other words, if a command is labeled, other semantically-similar commands will not be manually labeled but assigned with the same label. This approach works as we observed that many commands of the same user or different users are the same or semantically similar because i) users tend to use the same functions and ii) the language composition of commands is limited by the functions. We use a similarity score threshold of 0.88 to reduce the labeling effort. We observed that using thresholds much smaller than 0.88 results in many inconsistent cases. For example, in Table 2, two commands: *"Alexa I have the flu today when did I contract the flu virus"*, and *"Alexa when is the flu virus discovered."* have similarity score at 0.8616 but should be labelled differently. When using 0.88, the number of to-be-labeled commands is reduced to 3,667, which is considered an acceptable effort, and thus, we do not increase the similarity score threshold even higher. As a future work, an automatic approach to determine the threshold value *th* is to evaluate samples (randomly selected from the real dataset and labeled by humans) with a goal to keep minimum cases in which two commands (similarity score $\geq th$) have different labels.

Table 2: Extending labels using similarity score 0.88

Two commands	Similarity	Same
Alexa what time is the Gleneagle festival today. Alexa what time is the Gleneagle belt festival today	0.9891	✓
Alexa level four please Alexa level four	0.9658	✓
Alexa what is the weather in Bethesda. Alexa what is the weather.	0.8846	✓
Alexa I have the flu today when did I contract the flu virus. Alexa when is the flu virus discovered	0.8616	✗
Alexa what is the best time of day to take a multivitamin. Alexa is it better to take a multivitamin with food or without food	0.6601	✗

In Table 3, we show an example of 10 commands and their uniqueness scores. The uniqueness score of the first command is defined as 1. Then, starting from the second command, the uniqueness score is calculated based on the largest similarity score between the new command and the previous commands. For example, many "play music" commands are used here and similar, and thus their uniqueness score drops to <0.4. The 4th and 7th commands' uniqueness scores are high because "checking weather" and "set level" commands are used for the first time.

Table 3: Uniqueness of commands

i	c_i	max(sim)	uniqueness
c_1	Alexa play George gGershwin music	-	1.0
c_2	Alexa play Leonard Bernstein music	c_1	0.2100
c_3	Alexa play Carole King music	c_1	0.3132
c_4	Alexa what is the weather in Friendship Heights	c_3	0.5154
c_5	Alexa play classical music and turn it off	c_3	0.3772
c_6	Alexa play classical music	c_3	0.2613
c_7	Alexa level four	c_3	0.5265
c_8	Alexa level four	c_7	0.0
c_9	Alexa play Hawaiian music	c_8	0.2901
c_10	Alexa play George Gershwin music	c_1	0.0

B Discussion

Subjective opinions of privacy sensitivity. VPASS analyzes the privacy sensitivity of commands subjectively using annotators' knowledge and NLP models. Though we recruited five annotators ($25 incentive to each annotator) and used the majority votes, similar to the method in [10], the annotators' knowledge may not reflect the opinions of the VAS users. We realize that the annotators are young (about 20 years old) while the VAS users generating our dataset are much older ($\geq$ 65 years). It is impractical to request the VAS older adult users to annotate their own commands or find similar age groups to annotate the commands. To obtain general privacy knowledge of the voice commands, we will consider recruiting a larger number of annotators and more age-diverse groups to annotate the commands, which may help our model to extract accurate opinions on privacy sensitivity. Another direction is to develop a personalized model, which would require more training data and labels from the VAS users.

Linking to privacy surveys. In our project, we administered a privacy survey with 15 older adults every three months. We plan to correlate the usage of commands, the information disclosure analysis results, the privacy-sensitive analysis results, and the privacy survey results. This would help us understand whether their privacy opinions in the privacy survey are related to their actual VAS usage. Specifically, if their privacy concerns increase, they may have fewer VAS interactions, or they may generate less privacy-sensitive commands. Another interesting experiment is to have them finish a privacy survey after adopting VPASS. As VPASS allows users to effectively review VAS usage, their privacy concerns may be more accurate in the privacy survey.

Privacy intervention. VPASS passively monitors the commands downloaded from the VSP and provides useful insight back to the user via an intuitive interface. VPASS does not interfere with the real-time interaction between the users and the VAS. We envision VPASS may be more useful if it is integrated into the VAS such that any alert can be played in real-time to the users before the information is disclosed to the voice service provider. The users, after being alerted, can confirm the continued use of the privacy-sensitive commands. However, such intervention is difficult to implement and may affect the VAS use experience due to additional communication delay and effort, especially in a false positive case. Currently, we believe the privacy management of VAS usage is missing. VPASS is the first framework to enable users themselves to manage their privacy risks of VAS usage intuitively and effectively.