Preliminary Investigation of Predicting Next-use Mobile Apps Using App Semantic Representations

Cheng Chen, Takuya Maekawa, Amagata Daichi, Takahiro Hara
Osaka University
Osaka, Japan
{chen.cheng, maekawa, amagata.daichi, hara}@ist.osaka-u.ac.jp

***Abstract*—This study presents our preliminary investigation of predicting the next-use mobile Apps based on the App usage history of a target user that can facilitate the user to select an App from an entire list of Apps. The proposed method is designed to train a next-use App prediction model using the usage history of other users (source users) to cope with the cold-start problem of a system in which the training data from a target user is considered to be insufficient when the user begins to use the prediction system. We predict the usage of cold-start users and Apps by leveraging the semantic similarities between the Apps that are installed on the smartphones of the source users and the Apps that are installed on the smartphones of the target user.**

***Index Terms*—Mobile Apps, smartphones, next-use App**

I. Introduction

Because of the recent proliferation of smartphones, the number of available smartphone Apps in App stores is rapidly increasing. This huge number and diversity of Apps enable the installation of a large number of Apps on a user's smartphone. Although the large number of available Apps makes our lives more convenient, it also introduces a new challenge because selecting a particular App from all the installed Apps can be a time-consuming task. To assist a user in selecting Apps in an efficient manner, methods for predicting a next-use App that recommend probable candidates to the user have been studied in the mobile computing, pervasive computing, and recommender system research communities.

Mobile Apps are often used in conjunction with other relevant Apps [1]. For instance, when a user uses a smartphone for his or her business work, he or she may initially use a 'word processor' App. When he or she wants to send the edited document to others, the next-used App is likely to be an 'e-mail' App. Considering this feature, a next-use App can be predicted by supervised learning methods based on a user's App usage history.

However, supervised learning-based next-use App prediction poses several cold-start problems, the prediction model cannot be trained immediately after a user firstly installs the next-use App prediction system or install a new App from App store, since no usage history of the user and the App. To alleviate the aforementioned cold-start problems, some studies consider leveraging other users' (source users) usage histories to construct a prediction model for a target user. However, Apps that are installed on the smartphones of source users are different from those installed on the smartphones of the target user, making it difficult to recommend Apps that are not installed on the smartphones of source users to the target user (especially for newly released Apps). In our dataset, 26% of the installed Apps were categorized as unseen Apps.

To cope with the aforementioned problems, we propose an App recommendation method by combining multi-class classification and the semantic representations of a smartphone App. The proposed method permits us to use the source users' usage histories to train a prediction model tailored to a target user to alleviate the cold-start problems. Our concept is to obtain training data in the 1-of-K representations tailored to a target user from the source users' usage histories by using App semantic representations.

Let us assume that a series of App usages that is obtained from the source users is provided. We obtain a series of K-dimensional vectors in the 1-of-K representation tailored to the target user from the series of App usages, where K is the number of Apps that are installed on a target user's smartphone. We calculate the similarity between each pair of an App installed on a target user's smartphone and an App in the source users' usage histories based on their semantic representations and subsequently convert the App usage history of the source users into a series of the 1-of-K representation of a target user's App based on the calculated similarities. The series of K-dimensional vectors is further used to train a multi-class classifier (next-use App prediction model) for the target user.

II. Related work

A. Next-use App prediction

There have been some previous studies related to the next-use App prediction. Sun et al. [2] used a prediction model that utilized some App temporal features such as frequency and duration. Liao et al. [3] designed a time-based App predictor, extracting some features from the App usage trace such as an App's usage count in the entire usage trace, usage count in the temporal bucket, and the frequency of App usage. In contrast, we attempted to cope with the cold-start problems in the next-use App prediction. Further, our method, which is based on deep learning, does not require handcrafted features.

B. Cold-start problems in next-use App prediction

Few studies have considered the cold-start problems that have been mentioned in the introduction. Bazea-Yates et al. [4] divided the cold-start problems into App cold-start and user

978-1-5386-9151-9/19/$31.00

cold-start problems. For the App cold-start problem, where a user installed a new App on his or her smartphone, they used the App usage information obtained from other users for generating predictions. For the user cold-start problem, where a first-time user used the App recommendation system, they used a similar user's model to predict the behavior of the new user.

III. Method

A. Preliminaries

We use source users' usage histories to train a prediction model and then generate predictions for a target user. We define an App, source users' U, and target user's u_t as follows:

Definition 1 (App).
A set of Apps is installed on a user's smartphone, i.e., $\mathcal{A} = \{a_{\grave{1}}, a_{\grave{2}}, a_{\grave{3}}, ..., a_{\grave{n}}, ..., a_{\grave{K}}\}$, where $a_{\grave{n}}$ is the n^{th} App in the set. In contrast, the i^{th} used App in a user's usage history can be represented as a_i. When $a_{\grave{n}}$ is the i^{th} used App, a_i becomes equal to $a_{\grave{n}}$. The i^{th} used App a_i by a user is represented as a semantic vector v_{a_i}. The App is also represented as $o^K_{a_i}$ in accordance with the 1-of-K scheme, where K is the number of dimensions of the vector, i.e., the number of Apps installed on the user's smartphone. In addition, we refer to an App that is not installed on the source users' smartphones but is installed on the target user's smartphone as an " unseen App." Furthermore, we refer to an App that is installed on both the source users' smartphones and the target user's smartphone as an "existing App."

Definition 2 (Source users).
A group of N users $U = \{u_1, u_2, u_3, ..., u_N\}$. Each user u_i $(1 \leq i \leq N)$ has an App usage history with length M_i, i.e., $\mathcal{S}_i = \{a_1, a_2, a_3, ..., a_{M_i}\}$.

Definition 3 (Target user).
A user $u_t \notin U$, who has usage history with length M_t, i.e., $\mathcal{S}_t = \{a_1, a_2, a_3, ..., a_{M_t}\}$. When we wish to predict a next-use App, we do not use all the usage histories to generate predictions because an App that was used a long time ago exhibited little relation to the latent next-use App. In this study, we use a sequence of App usage histories with lengths of k $(1 \leq k \leq M)$, i.e., $s = \{a_{i-k}, a_{i-k+1}, ..., a_{i-2}, a_{i-1}\}$, to predict a next-use App a_i. Further, the next-use App prediction problem can be defined as follows:

Definition 4 (App prediction).
Given a series of App usage histories with length k $(1 \leq k \leq M)$, i.e., $s = \{a_{u_t,i-k}, a_{u_t,i-k+1}, ..., a_{u_t,i-2}, a_{u_t,i-1}\}$, of a target user u_t, the probability that each App $a_{\grave{n}}$ that is installed on this user's smartphone will be a next-use App, i.e., $P(a_{\grave{n}}|s)$, is calculated. We further select the top-N Apps to be the next-use App candidates.

B. Overview

An overview of our proposed method is depicted in Figure 1. The proposed method has two phases, training and prediction. In the training phase, we initially compute an App semantic representation, i.e., an App vector, for each App installed on the source users' or target user's smartphones. Further, we utilize the sequences of App semantic vectors from the source users' usage histories to generate training data that are tailored to the target user by leveraging the App semantics. Subsequently, we train a prediction model for the target user based on a deep neural network comprising long short-term memory (LSTM) on the tailored training data. In the prediction phase, we use the App usage history of the user through $i-1$ to predict the target user's i^{th} App usage candidates..

C. Semantic representation of App

We assume that we obtain a description of each App from the App store, which is used to build a semantic App vector. An App description describes the features of an App and the functions that are to be provided to users, similar Apps have similar description. Therefore, we construct semantic App vectors from descriptions to calculate the similarity between various Apps. Note that we cannot obtain the descriptions of Apps that are pre-installed or directly installed from the APK files. The titles of such Apps are used in place of descriptions.

In our proposed method, we first select W keywords of an App that represent the App well from the extracted words using the importance of the words that are computed based on tf-idf. Further, we use a word embedding model word2vec to obtain a word vector for each keyword representing the semantics of the keyword. We use an external pre-trained word2vec model on Japanese Wikipedia[1] to compute a word vector for each keyword. Finally, we compute the mean vector of the W keywords and regard it as the App's semantic vector. We ignore the keywords that are not present in the external Japanese Wikipedia corpus. We use this method to build an App semantic vector for each App on the source users' or a target user's smartphones.

Figure 2 depicts the semantic vectors of 9 Apps from 3 categories that are constructed using the aforementioned method that is projected into two-dimensional space using t-SNE. Same color Apps belong to same categories. We can observe that approximately all of the Apps belonging to the same categories are grouped together and that the distance between Apps belonging to different categories is larger than that for those in the same category.

D. Generating training data tailored to a target user

In order to predict both existing Apps and unseen Apps, we design an algorithm generating the training data is tailored to a target user. Firstly, we initially prepare the conversion matrices that compute the cosine similarity between all Apps installed on target user and source users' smartphone. Next, using the matrices, we probabilistically generate multiple sequences of App usages tailored to the target user from the sequence of App usages of the source user. In detail, we probabilistically construct a convert table that describes a mapping from an App of the source user to an App of the target user. Because a

[1] http://www.cl.ecei.tohoku.ac.jp/~m-suzuki/jawiki_vector/

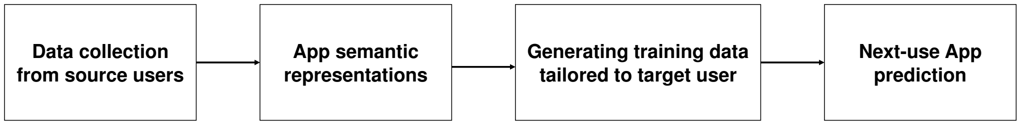

Fig. 1. Overview of the proposed method

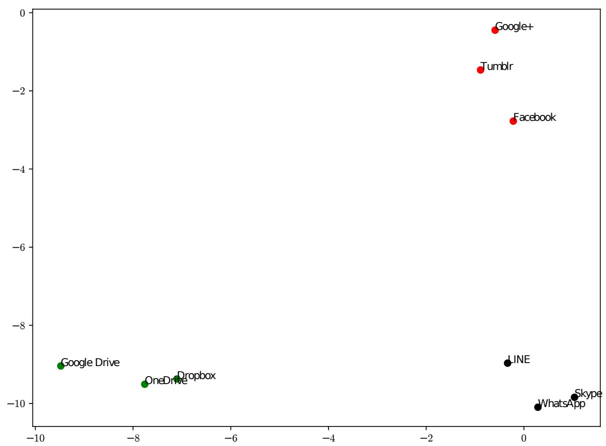

Fig. 2. Visualization of the App semantic vectors

mapping from an App of the source user to an App of the target user that is computed from the App semantic representation is not entirely accurate, we mapping the source users' Apps and target user's App using the roulette-wheel-selection that means high similarity Apps have high probability to be mapping. We convert all source user's App into similar target user's App according to the computed similarity in convert matrices. Lastly, using the convert table, we convert the sequence of App usage by a source user into a training sequence tailored to the target user so that all Apps in training data are from the target user.

E. Predicting the next-use App using a neural network

We train a neural network to output the $k+1^{th}$ used App when a sequence of usages from 1 to k is provided as input. Note that each App usage is represented in the 1-of-K scheme with the number of dimensions being the number of Apps installed on the target user's smartphone. Because the App usage history is the time-series data, we choose the LSTM model to generate predictions. In accordance with the data structure of the training data, we adopt a two-layer many-to-one multilabel classification LSTM model whose input and output are a sequence and a fixed-size vector, respectively. In our case, the input of the network is a series of App usage histories represented in the 1-of-K scheme, and the output is a vector consisting of the class probability of each App. The network comprises two LSTM layers with 300 nodes using a rectified linear units (ReLU) activation function and an output softmax layer. To reduce overfitting, we use dropout, a simple regularization technique where randomly selected nodes are ignored during training. We train the network using backpropagation based on Adam to minimize the categorical cross-entropy between estimates and the ground truth. In the prediction phase, an App usage sequence with length k obtained from the target user is provided as input to the network, and the network outputs the probability with which each App of the target user will be a next-use App. Finally, the Apps with the top-N probabilities are chosen as the next-use App candidates.

TABLE I
STATISTICS OF OUR DATASET

	total # of Apps	total # of usage sequences
Unseen Apps	383	14,052
Existing Apps	1,064	174,549
All Apps	1,447	188,601

IV. EVALUATION

A. Dataset

We collected an App usage dataset from 20 participants using our developed Android App that was installed on their smartphones. Each participant in the dataset has 9,430 usage history data items on an average collected for approximately one hundred days. To obtain the App semantic representations, we retrieved a description of each App from Google Play. For Apps that were not available on Google Play, we used the titles of the Apps instead of descriptions. Table I presents the total number of unseen Apps and existing Apps and their usage sequences. The usage sequence of an unseen App is the usage sequence with length k whose next-use App, i.e., answer, is an unseen App ($k=5$). In contrast, the usage sequence of an existing App is the usage sequence whose next-use App, i.e., answer, is an existing App.

B. Evaluation methodology

1) Evaluation measure: To evaluate the App prediction methods, we employ a top-N prediction accuracy metric in our experiment. If the *any* App in a set of N candidate Apps is actually the next-use App, we regard the next-use App to be accurately predicted. The prediction accuracy metric can be defined as **Accuracy@N** $= \frac{\sum_{i=1}^{T} H_i^N}{\sum_{i=1}^{T} A_i}$, where H_i^N is the number of accurately predicted next-use Apps for a test user i when N candidates are provided, A_i is the total number of test data for test user i, and T is the number of test users.

2) Methods: We evaluate the following methods.

• **MFU** (most frequently used): The top-N candidates are the most frequently used N Apps in the training data.

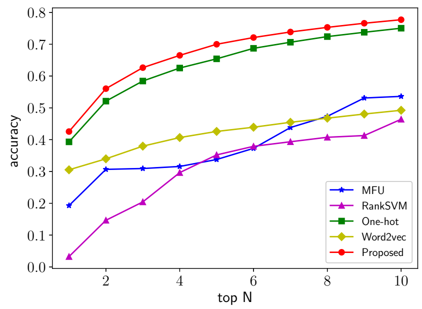

Fig. 3. Transitions of Accuracy@N when N is varied

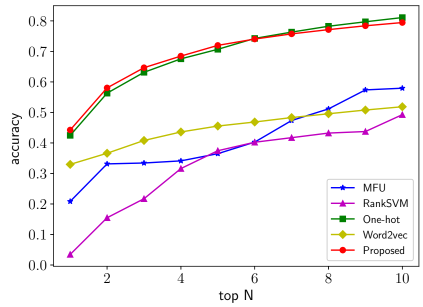

Fig. 4. Transitions of Accuracy@N for existing Apps when N is varied

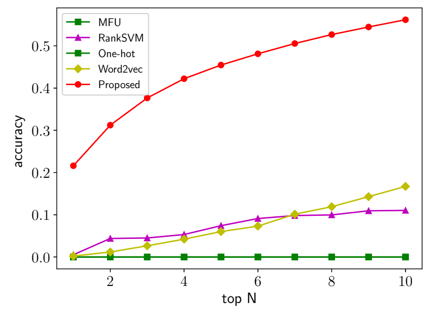

Fig. 5. Transitions of Accuracy@N for unseen Apps when N is varied

• **RankSVM**: A ranking method proposed in [5], which is a learning-to-rank algorithm for query-based document search, is used to obtain the top-N next-use Apps.
• **One-hot**: The neural network architecture is identical to that of the proposed method. However, the training data are not tailored to a target user. Each App is represented in the 1-of-K scheme where K is the size of the set of all Apps installed on the smartphones of source/target users.
• **Word2vec**: The neural network architecture is similar to that of the proposed method. Each App that is used in the input and output of the network is represented by a semantic App vector. In the prediction phase, we calculate the cosine similarity between an output vector and a semantic vector for each of the Apps installed on a target user's smartphone. The top-N candidates are the N Apps of the target user with the top-N cosine similarities.
• **Proposed**: This is the proposed method.

We use "leave-one-participant-out" cross validation to evaluate the aforementioned methods. Therefore, we consider one participant to be a target user and the remaining participants as source users.

C. Results

Proposed achieved the optimal performance and outperformed One-hot by approximately 3%. Figure 4 depicts the Accuracy@N of the five methods for the existing Apps. The figure exhibits that the performances of Proposed and One-hot are nearly identical. This is because the neural network architectures of these methods are identical. In contrast, as depicted in Figure 5, with respect to the Accuracy@N of unseen Apps, Proposed considerably outperformed the other methods. Surprisingly, Proposed achieved 56% accuracy when $N = 10$ even though the usage history of these unseen Apps by source users is not available at all. One-hot and MFU could not predict the use of unseen Apps at all because of their architectures. While their accuracies are poor, RankSVM and Word2vec could also predict the usage of these unseen Apps. This is because these methods compute the App candidates based on the semantics of Apps. However, Proposed, the architecture based on the 1-of-K scheme, works effectively for this multi-class classification task.

V. Conclusion

This study proposed a new method for predicting the next-use mobile Apps based on the App usage history of a target user using the training data collected from other users (source users). The proposed method makes use of the semantic representations of Apps to predict the usage of Apps that are not installed on the smartphones of source users by a target user. Our experiment that was conducted using the actual App usage data revealed that the proposed method achieved an accuracy of 56% (Accuracy@N; $N = 10$) while predicting the usage of Apps that were not installed on the smartphones of source users.

Acknowledgment

This research is partially supported by JSPS Grant-in-Aid for Scientific Research (A) Grant Number JP26240013 and JST CREST Grant Number J181401085.

References

[1] N. Natarajan, D. Shin, and I. S. Dhillon, "Which app will you use next?: Collaborative filtering with interactional context," in *the 7th ACM Conference on Recommender Systems (RecSys '13)*, 2013, pp. 201–208.
[2] C. Sun, J. Zheng, H. Yao, Y. Wang, and D. F. Hsu, "Apprush: Using dynamic shortcuts to facilitate application launching on mobile devices," in *ANT/SEIT*, 2013.
[3] Z.-X. Liao, Y.-C. Pan, W.-C. Peng, and P.-R. Lei, "On mining mobile apps usage behavior for predicting apps usage in smartphones," in *the 22Nd ACM International Conference on Information & Knowledge Management (CIKM '13)*, 2013, pp. 609–618.
[4] R. Baeza-Yates, D. Jiang, F. Silvestri, and B. Harrison, "Predicting the next app that you are going to use," in *the Eighth ACM International Conference on Web Search and Data Mining (WSDM '15)*, 2015, pp. 285–294.
[5] T. Joachims, "Optimizing search engines using clickthrough data," in *the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)*, 2002, pp. 133–142.