Estimating User Contexts from Mobile Application Usage Histories

Toshimitsu Kamiya, Tatsuya Nakamura, Takuya Maekawa, Daichi Amagata, Takahiro Hara
Graduate School of Information Science and Technology, Osaka University, Osaka, Japan
Email:{kamiya.toshimitsu, nakamura.tatsuya, maekawa, amagata.daichi, hara}@ist.osaka-u.ac.jp

***Abstract*—While context information of a smartphone user such as his/her current location, the number of companions, mental conditions, and health conditions can be fundamental information towards smarter services, it is not easy to estimate these information with built in smartphone sensors. This paper develops a user context estimation model based on mobile application usage histories. As user application history and context are assumed to be time series data that change over time, a recurrent neural network was employed to learn temporal relationship between the history and context. The experimental results confirmed that the proposed model was effectively able to estimate user context from application usage history.**

***Index Terms*—context; recurrent neural network; smartphone; application**

I. Introduction

A. Background and Motivation

Smartphones now have high performance OS such as android, high performing CPUs, and sensors such as acceleration and GPS, enabling us to develop high functional mobile applications.

Smartphone users use mobile applications depending on their activities, locations, companions, moods, and conditions, for example, when a user is traveling, they may use a map application to browse surrounding for sightseeing spots. Mobile applications provide various services, for example, restaurant search applications utilize location information from the GPS sensors to search for restaurants near the user's current location. Therefore, there are distinct relationships between the users' situations and their application usage patterns. By revealing these relationships, we can improve service of mobile applications [2]–[4], [7], and recommend mobile applications which are suited to the user's situation at the time [8], [13], [19].

There are two types of information that describe user situations: (1) information from physical sensors, and (2) contexts. Smartphones have several physical sensors, such as GPS sensors, accelerometers, and gyroscopes, from which it is possible to know smartphone user situations such as location or activity. In contrast, context information describes the semantic and subjective situations of the smartphone users, such as places (semantics of location), activities, companions, and user mood. Previous studies have demonstrated that these contexts contain valuable information for mining user behavior and predicting the next application use [14], and estimating the current symptoms of depressed patients [15].

While the previous studies have demonstrated the effectiveness of these contexts, it is difficult to identify the contexts from the physical sensors. Therefore, the previous studies have developed mobile applications that directly ask users about their contexts at the time. However, this is often annoying for users and can reduce user participation. Therefore, context estimation techniques that do not place large burdens on users are highly demanded. However, as mentioned, estimating user contexts such as subjective information from solely sensory data is difficult.

In this paper, we address the problem of estimating smartphone user contexts. To estimate these contexts, we consider to utilize mobile application usage histories that can be automatically collected and mobile application semantic information such as the app descriptions in the app stores. Therefore, the problem can be defined as follows: given a smartphone user mobile application usage history and a context type to be estimated, we predict whether the user is in a given context at the time of the latest application use in their history. The problem has the following challenges.

1) **There are implicit relationships between application usage history and context:** To estimate the user contexts, we need to consider not only the specific applications of a context, but also the temporal application usages patterns. However, as there is no explicit knowledge about these relationships, it is necessary to extract the relationships from the user application usage histories automatically.
2) **Smartphone users often install new mobile applications:** Many new applications are released every day; therefore, even if a user is in the same context, the application usage changes over time, which makes estimation difficult because the user has never used a newly installed application in any context before.

To handle these newly installed applications, it is necessary to find previously used and functionally similar applications. Smartphone users tend to use functionally similar applications in the same context. For example, when a user installs a new restaurant search application and has previously used other restaurant search applications, it is highly possible that the newly installed application is to be used in the same situation as the existing applications; however, measuring these application similarities can be difficult.

978-1-5386-9151-9/19/$31.00 ©2019 IEEE

B. Contributions

In this paper, we address the problem of estimating smartphone user contexts using mobile application usage histories. We propose a method that builds estimation models by learning the relationships between the application usage histories and the contexts. The proposed estimation models employed a LSTM (Long Short-Term Memory) [6] RNN model, which are suited to time series data estimation problems. In addition, to measure the similarities between mobile applications, the mobile applications are represented as continuous vectors using their respective descriptions from the app store. The descriptions contain specific terms which represent features and functions of mobile applications, and it is available even when the application is newly issued. The proposed method extracts such terms and convert them into the continuous vector of a mobile application by using a pre-trained word2vec model [16]. As a result, functionally similar applications are represented as similar vectors, and the proposed method is able to estimate the contexts for newly installed applications based on past information from other similar applications.

II. Related work

In this section, we first describe previous study on user context estimation using sensor information and smartphone usage history, after which we examine the previous study on context estimations using RNN.

A. Smartphone user context estimations

Previous studies have sought to estimate user contexts from smartphone history data and data collected from built in microphones and sensors. Sano and Picard [12] proposed a method to estimate user stress using smartphone usage history such as the number and length of calls and short messages (SMS) as well as acceleration and skin conductance collected from sensors attached to the user's wrist. Lu et al. [10] constructed a model that estimated the degree of user stress from features such as the pitch of the user's voice, which were collected from built in smartphone microphones, and found that even though higher performance was achieved with a greater amount of collected data, there was a psychological burden to users as the data collection increased. Bogomolov et al. [1] proposed a method to estimate user stress based on user characteristics, weather information, and user smartphone history such as the number and length of calls and SMS, and confirmed that all these factors were useful in estimating user stress levels. LiKamWa et al. [9] constructed a model to estimate user moods based on Russell's circular model using call history, SMS history, and the number of application launches and use periods. To improve performance for users with small amount of data, they also proposed a method that estimated user mood using other user's data.

These studies collected data from built in smartphone sensors and user smartphone histories to estimate user context. However, the data collected from sensors such as the microphone and GPS also include privacy information and therefore the continuous collection of these data was found to be a psychological burden to the user. Therefore, in this study, we estimate user context using user application history as this can be easily collected and does not include the private information included in call and SMS histories, thereby reducing the psychological burden. Moreover, previous study the context estimation was determined only from the static relationships observed between the smartphone histories and contexts. However, as user smartphone histories and contexts are seen as only one type of time series data that changes over time, in this study, an estimation model is constructed that harnesses the power of the RNN to handle the time series data.

B. Context estimation using a RNN

RNN have had significant time series data analysis achievements. Vu et al. [18] proposed an RNN-based method to estimate user movements such as walking, running, and riding in a car using the built in acceleration, magnetic, and gyro sensors. Suhara et al. [15] proposed a model to estimate the moods and medication of depressed patients using support vector machine (SVM) [17] and LSTM [6], which is an RNN models, from which it was confirmed that the LSTM was able to capture the changes in the patients' long-term conditions.

These previous studies have confirmed that the RNN can effectively estimate user contexts from time series data such as smartphone sensors and usage history. Therefore, in this paper, we also employ an RNN estimation model to estimate user contexts only from user application history.

III. Proposed method

A. Outline of the proposed method

In this study, the user context estimation problem is seen as a binary classification problem. Using the mobile application usage history and the context type, we seek to predict the given context from the latest application use. To solve this problem, an RNN-based estimation model is constructed that estimates user contexts from the mobile application use history.

Figure 1 shows the procedure for the proposed method. First, the proposed method converts an application in the application history into a continuous vector for input into the estimation model. Then, the estimation model, which has been trained in advance on the user application history, judges whether the user context when the input application was last used is the estimation target. In the following subsections, we introduce the proposed method in detail.

B. Vector representation of the user application history

For the proposed method, it is necessary to convert the user application history into a vector format that can be input into the context estimation model (neural network). In this study, two vector representation formats; One-hot vector, and semantic vector are generated from the application descriptions to be used as the vector representation formats. Each vector representation format is described below.

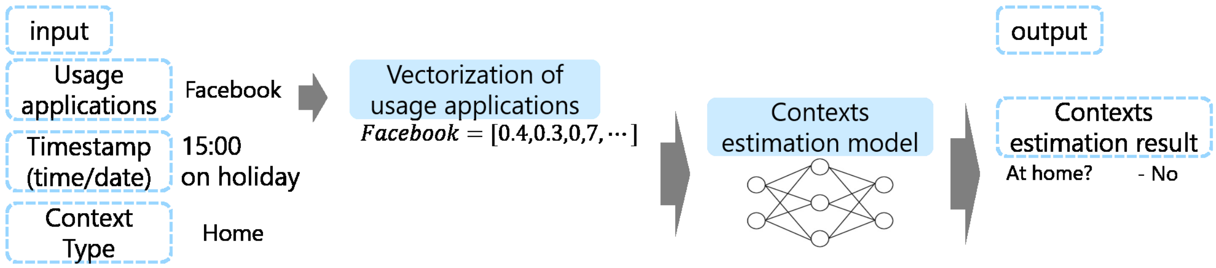

Fig. 1. Procedure for the proposed method

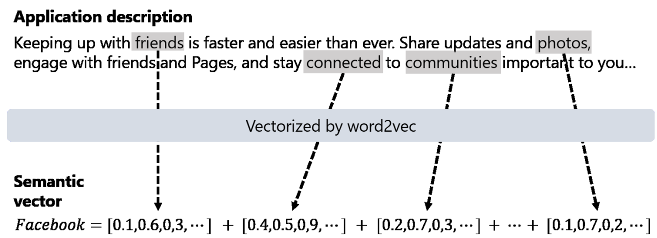

Fig. 2. Semantic vector creation example using application descriptions (Facebook)

1) One-hot vector representation: The simplest vector representation is the One-hot representation. The One-hot vector is an N-dimensional vector, where N is the total number of applications in the history, and each dimension corresponds to one application. With a certain application, the vector is given a dimension weight of 1, with the weight of the other dimension being 0.

One-hot vectors are able to represent explicit information about application use; however, One-hot can not represent relations between applications. So, it has some problems estimating user context. First, when a user installs a new application, it is difficult to estimate the context because there is no context information from the past. Second, the new application can not be represented as a previous One-hot vector because there are no dimensions that correspond to the new application. To resolve these problems, we propose a semantic vector representation for mobile applications that uses the application descriptions from the app stores.

2) Semantic vector representation: Semantic vector representations use app store description, with all applications being seen as continuous vectors in a common semantic space. Therefore, if the apps have similar descriptions, their vectors are similar vectors. Even if a user installs a new application, it is possible to estimate user context using the context information from the use information for similar applications.

The proposed method generates semantic vectors as follows: (1) Given an application, keywords are selected from its description. To extract the keywords, TF-IDF scores for each word in the description are calculated, and the top ten words with the highest scores selected as the keywords. (2) Each keyword is then converted into a continuous vector using word embedding techniques, such as word2vec [11]. In the proposed method, a pre-trained word2vec model that adopted skip-gram with negative sampling and trained on the Japanese version of Wikipedia[1] [16] is used. In this model, each word was represented as a 200-dimention vector. (3) All vectors were summed into one vector, and then used as the semantic vector for the given application. Figure 2 shows the procedure for generating a semantic vector for an application.

C. Context estimation model using RNN

In this section, we first outline RNN and LSTM used as the context estimation model in the proposed method, after which we describe the context estimation model in detail.

1) RNN and LSTM: RNN is a generic term for neural networks with internal closed loops that can store past information internally and capture past to the present time series data features, RNN have been used for signal processing and natural language processing and other areas and have shown outstanding performances.

While it is possible to estimate the RNNs that capture the characteristics of past information, when learning long-term data, past information is limited by a gradient elimination problem. The LSTM solves this problem by replacing the RNN intermediate layer unit with a memory unit that can efficiently control information using three gates; an input gate, an output gate, and a forgetting gate. Based on this internal structure, the LSTM can handle long-term time series data much better than a naive RNN. In our proposed method, LSTM is used in the estimation model to assess the long-term relationships between user application history and user contexts.

2) Context estimation model network structure: Figure 3 shows the network structure for the proposed model when the semantic vector is used. The numbers in the figure represent the output dimensions for each layer, and the characters between each layer represent the activation functions used in the preceding layer. In the proposed model, the application vector and the time and day of the week when the application was launched are taken as the inputs. First, the model converts the application vector and usage time into a 16 dimensional vector and a 2 dimensional vector in the dense layer, after

[1] https://github.com/singletongue/WikiEntVec

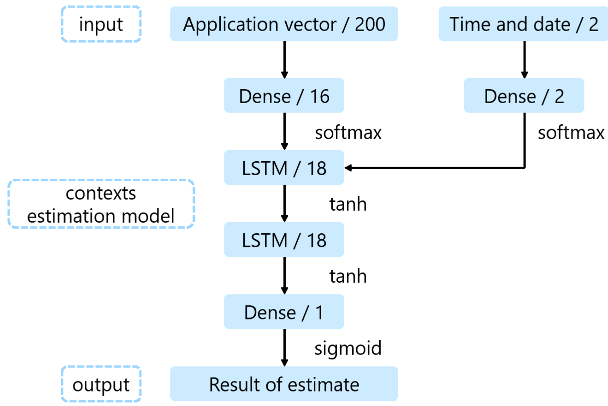

Fig. 3. Network constitution for the context estimation mode (semantic vector)

TABLE I
CONTEXT LIST TO ESTIMATE

Context Category	Context type
Place	Home
	School / Workplace
	Move
	Outdoor
Companions(num)	With someone
Mind	Good
Health	Good

which it concatenates the data from both vectors into the 18 dimensional vector and inputs the vector into the two LSTM layers. Finally, the model outputs the probability as to whether the user was in the target context when the application was being used.

IV. EVALUATION

A. Setting

1) Dataset: Application usage history and context datasets were collected using a developed android application that was installed on their smartphones from February 9, 2015 to December 10, 2015. We used data of five participants with large number of data for experiments. This developed application observed the user application usage histories; that is, application launches; and recorded the application IDs, the application usage timestamps, and the user contexts; place, number of companions, mind, and health. Table I gives the list of contexts to be estimated. As each context had an effective time, the application being used within this time was linked with the context. Overall, the dataset had 211,931 application usage history data entries, with the average number of application usage histories per user being 520.1 and the median value being 50. From the logs from all application use logs, the number of logs associated with even one context within the effective time was 14,043 (6.6%). In this experiment, the data from each user was divided into 80% for the first half of the period and 20% for the second half of the period. TableII gives the statistical information for the data that was used in this experiment.

TABLE II
STATISTICS FOR THE DATA SET USED FOR EVALUATION

Context Type	User	Training data		Test data	
		Positive	Negative	Positive	Negative
Home	A	355	2,196	82	556
	B	616	336	193	46
	C	1,186	938	323	209
	D	206	195	25	76
	E	1,510	1,350	581	135
School / Workplace	A	1,742	809	342	296
	B	198	754	36	203
	C	351	1,773	146	386
	D	99	302	34	67
	E	323	2,537	0	716
Move	A	297	2,254	99	539
	B	138	814	10	229
	C	312	1,812	63	469
	D	39	362	0	101
	E	1,027	1,833	135	581
Outdoor	A	157	2,394	76	562
	B	0	952	0	239
	C	275	1,849	0	532
	D	57	344	42	59
	E	0	2,860	0	716
Companions(num)	A	630	1,338	190	303
	B	575	165	185	0
	C	349	1,579	38	445
	D	53	278	6	77
	E	1,253	1,054	442	135
Mind	A	2,153	503	554	110
	B	257	808	0	267
	C	2,236	0	560	0
	D	444	20	117	0
	E	1,832	1,068	609	116
Health	A	2,019	737	664	25
	B	217	840	0	265
	C	2,190	51	561	0
	D	446	27	119	0
	E	1,078	1,824	154	572

2) Metrics: From Table II, it can be seen that the data set was imbalanced as the data entries were either skewed to a positive example or a negative example depending on the user and context type. Therefore, if a simple accuracy were applied, it would not be possible to properly evaluate the model performance as apparent high accuracy could be achieved by continuously outputting labels that had a larger number of data entries from the positive and negative examples [5].

In this experiment, the macro average f-measure was used as the evaluation metrics, which gives a score between 0 and 1; the closer the value is to 1, the better the performance of the estimation model. The macro average f-measure ($F1$-$score$) was defined using the following equation.

$$F1\text{-}score = \frac{1}{2}(F1_p\text{-}score + F1_n\text{-}score) \quad (1)$$

where, $F1_p$-$score$ and $F1_n$-$score$ are respectively the $F1$-$score$ for positive example and the $F1$-$score$ negative example, which were defined using the following formulas.

$$F1_p\text{-}score = \frac{2recall_p \cdot precision_p}{recall_p + precision_p} \quad (2)$$

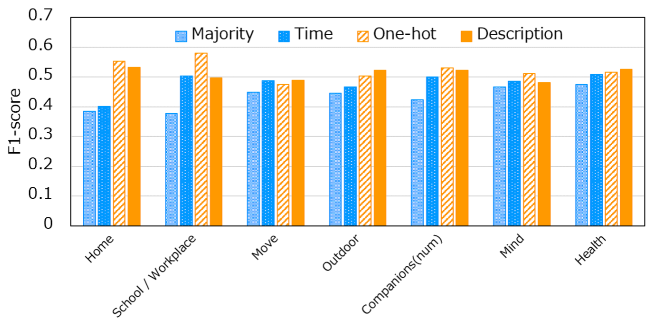

Fig. 4. Average estimation $F1$-*score* of the entire user

$$F1_n\text{-}score = \frac{2recall_n \cdot precision_n}{recall_n + precision_n} \tag{3}$$

where, $precision_{p,n}$ and $recall_{p,n}$ are the precision and the recall for the positive examples and the negative examples. Based on the above definition, the macro average f-measure overcame the imbalanced data classification problem as it took the maximum value 1 only when both the positive example and the negative example could be properly estimated.

3) Comparison method: In this experiment, the following context estimation models were compared.

- Majority: A model that always outputs a label with a larger number of data.
- Time: A model that estimates context using only time information (time, day of the week). The network structure of the model is determined by removing the application vector from Figure 3.
- One-hot: A model using the One-hot vector as the application vector in the context estimation model. Note that, as there were 259 applications in the dataset, the One-hot vectors had 259 dimensions in this experiment.
- Description: A model using the semantic vector generated from the application description as the application vector in the context estimation model.

B. Results and discussion

Figure 4 shows the results for the user context estimation. Compared to the comparison methods (Majority and Time), the proposed models (One-hot and Description) that used application usage history achieved good performances for each context. It suggested that utilizing application usage history is effective when seeking to estimate user contexts. Of the proposed models, the One-hot had a better performance than the Description for the contexts of "home", "school / workplace", "companion," and "mind". As the One-hot can explicitly represent information, it is considered that the same application is more likely to be used for contexts strongly related to life such as " home " and " school / workplace " , and the One-hot model had higher performance. As the Description model was represented by a vector that assessed use for similar applications as well, it was difficult to capture the detailed usage for the different applications. However, in contexts that did not represent a specific location such as "move" or "outdoor", the Description model had higher performance because the user was more likely to use different applications than one specific application. Therefore, it was assumed that the One-hot model was suitable for contexts with fixed application usage patterns, and the Description model was suitable for contexts in which different (but similar) applications were used. To further verify these it, we used the Jaccard coefficient to investigate user application changes between the training and test periods. The Jaccard coefficient represents the similarity between two sets X and Y, and is defined as follows.

TABLE III
USER AVERAGE OF JACCARD COEFFICIENTS IN EACH CONTEXT

Context	Jaccard coefficients	
	Positive example	Negative example
Home	0.440	0.426
School / Workplace	0.414	0.417
Move	0.289	0.465
Outdoor	0.149	0.446
Companion(num)	0.492	0.349
Mind	0.370	0.257
Health	0.349	0.222

$$Jaccard_{X,Y} = \frac{|X \cap Y|}{|X \cup Y|} \tag{4}$$

where X and Y are respectively sets of unique applications in the training data and the test data.

Table III shows the Jaccard coefficients between the training data and the test data. From Figure 4 and Table III, it can be seen that the contexts in which the One-hot achieved good performances tended to have larger Jaccard coefficients for the positive and negative examples. In other words, these contexts achieved higher performance if the user application context changed little over time and the applications in that context had been learned. As the One-hot model was able to explicitly express the information in the used application, it was able to achieve higher performance than the Description for contexts with these features.

The contexts in which the Description achieved good performances tended to have smaller Jaccard coefficients, which indicated that the user application usage changed over time; i.e., the user tended to use new applications in the test period. One-hot was not able to work well in these situations because there was no information about new applications in the training period. However, the Description was able to estimate the contexts for the new application usage by utilizing information from similar applications in the training period. Therefore, the Description achieved good performances in these contexts.

In practice, the Description was more feasible for real life situations than the One-hot because there are many new applications issued every day in the app stores. However, with One-hot, it is necessary to determine the number of vector dimensions (i.e., the number of applications) before training the estimation model. Therefore, every time a new application is issued, it is necessary to redefine the vectors to support the new application as an input, and then retrain the model with the redefined vectors. However, the Description model represents both the new and existing applications as vectors in a common semantic space by utilizing the application descriptions.

V. Conclusion

In this study, we proposed an RNN-based model that used binary classification to determine whether a user context was the target context when the application was being used, with the application being used as the input. In the proposed models, application vectorization was assessed using two methods. The One-hot vector representation was based on application ID information and the semantic vector representation was based on application descriptions. LSTM was then used in the estimation model to consider the temporal relationships between application use and context.

User context estimation experiments were conducted using data collected by a developed android application. The experimental results confirmed that the proposed method was able to estimate user contexts more accurately than comparison methods that do not use application usage history data. We then compared the performances of the proposed methods and found that the One-hot vector representation was able to effectively estimate contexts when the applications did not change over time, and that the semantic vector representation was able to effectively estimate contexts when user applications did change over time.

In future work, we plan to improve the estimation performance using information collected by sensors built in smartphone and improve context estimation using information from other users for users with small application histories.

Acknowledgment

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

References

[1] A. Bogomolov, B. Lepri, M. Ferron, F. Pianesi, and A. S. Pentland, "Daily stress recognition from mobile phone data, weather conditions and individual traits," in *MM*, 2014, pp. 477–486.

[2] K. Church, D. Ferreira, N. Banovic, and K. Lyons, "Understanding the challenges of mobile phone usage data," in *MobileHCI*, 2015, pp. 504–514.

[3] T. M. T. Do, J. Blom, and D. Gatica-Perez, "Smartphone usage in the wild: A large-scale analysis of applications and context," in *ICMI*, 2011, pp. 353–360.

[4] D. Ferreira, J. Goncalves, V. Kostakos, L. Barkhuus, and A. K. Dey, "Contextual experience sampling of mobile application micro-usage," in *MobileHCI*, 2014, pp. 91–100.

[5] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowledge and Data Engineering*, pp. 1263–1284, 2009.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] S. L. Jones, D. Ferreira, S. Hosio, J. Goncalves, and V. Kostakos, "Revisitation analysis of smartphone app use," in *Ubicomp*, 2015, pp. 1197–1208.

[8] 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 *CIKM*, 2013, pp. 609–618.

[9] R. LiKamWa, Y. Liu, N. D. Lane, and L. Zhong, "Moodscope: Building a mood sensor from smartphone usage patterns," in *MobiSys*, 2013, pp. 389–402.

[10] H. Lu, D. Frauendorfer, M. Rabbi, M. S. Mast, G. T. Chittaranjan, A. T. Campbell, D. Gatica-Perez, and T. Choudhury, "Stresssense: Detecting stress in unconstrained acoustic environments using smartphones," in *Ubicomp*, 2012, pp. 351–360.

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013, pp. 3111–3119.

[12] A. Sano and R. W. Picard, "Stress recognition using wearable sensors and mobile phones," in *AICC*, 2013, pp. 671–676.

[13] C. Shin, J.-H. Hong, and A. K. Dey, "Understanding and prediction of mobile application usage for smart phones," in *Ubicomp*, 2012, pp. 173–182.

[14] V. Srinivasan, S. Moghaddam, A. Mukherji, K. K. Rachuri, C. Xu, and E. M. Tapia, "Mobileminer: Mining your frequent patterns on your phone," in *Ubicomp*, 2014, pp. 389–400.

[15] Y. Suhara, Y. Xu, and A. Pentland, "Deepmood: Forecasting depressed mood based on self-reported histories via recurrent neural networks," in *WWW*, 2017, pp. 715–724.

[16] M. Suzuki, K. Matsuda, S. Sekine, N. Okazaki, and K. Inui, "A joint neural model for fine-grained named entity classification of wikipedia articles," *IEICE Transactions on Information and Systems*, vol. 101, no. 1, pp. 73–81, 2018.

[17] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *Journal of machine learning research*, pp. 1453–1484, 2005.

[18] T. H. Vu, L. Dung, and J.-C. Wang, "Transportation mode detection on mobile devices using recurrent nets," in *MM*, 2016, pp. 392–396.

[19] X. Zou, W. Zhang, S. Li, and G. Pan, "Prophet: What app you wish to use next," in *Ubicomp*, 2013, pp. 167–170.