Personalized Natural Language Understanding

Xiaohu Liu, Ruhi Sarikaya, Liang Zhao, Yong Ni, Yi-Cheng Pan

Microsoft Corporation, Redmond, WA 98052

{derekliu, ruhi.sarikaya, leozhao, yoni, ycpani}@microsoft.com

Abstract

Natural language understanding (NLU) is one of the critical components of dialog systems. Its aim is to extract semantic meaning from typed text input or the spoken text coming out of the speech recognizer. Traditionally, NLU systems are built in a user-independent fashion, where the system behavior does not adapt to the user. However, personal information can be very useful for language understanding tasks, if it is made available to the system. With personal digital assistant (PDA) systems, many forms of personal data are readily available for the NLU systems to make the models and the system more personal. In this paper, we propose a method to personalize language understanding models by making use of the personal data with privacy respected and protected. We report experiments on two domains for intent classification and slot tagging, where we achieve significant accuracy improvements compared to the baseline models that are trained in a user independent manner.
Index Terms: natural language understanding, personalization, user-dependent modeling

1. Introduction

Personal digital assistants (PDAs) have been receiving tremendous attention in the last couple of years as a means to enable information access, task completion, and ultimately to improve user's productivity [1]. The dialog system on PDAs consists of multiple components such as a speech recognizer, language understanding models, dialog management and language generation modules. The speech recognizer is responsible for converting speech into text. Language understanding models are used to extract semantic meaning from text inputs. The dialog manager adds additional contextual and back-end knowledge signals to the semantic analysis, ranks the hypotheses, and applies the dialog policy to generate a system action, returning information in the form of answer cards, links, or natural language responses from the language generation module [2, 3, 4].

The natural language understanding (NLU) module is responsible for semantically parsing the query to represent the query's domain, intents and semantic slots [5, 6, 7]. All of this information is often encapsulated and represented as a semantic frame. The domain classification model determines which domain (i.e. scenario or task) the query belongs to at a high level, such as *communication, weather, places, calendar, etc.* The intent model determines the specific intent of the user query in the detected domain, such as *send_email, call_contact, check_weather, find_place, get_directions, add_appointment, etc.* The slot tagger extracts the slots and entities contained in the query, such as *person_name, date, location, application_name, etc.*

Traditionally, NLU systems are designed to exhibit a fixed user experience, without adapting their behavior to the specific user using the system. This may not lead to desired user experience. A better design requires NLU systems know who their users are and adapt their behaviors by making use of the user's data. Knowing more information about each user helps the system understand their users better and serve them improved experience.

There are many signals available (e.g. personal contacts, installed applications, user's location, to-do list, Facebook posts, flights, user interests, music entity search, calendar snapshot, driving mode, cuisine search, etc.) to leverage and personalize the system accordingly.

For example, if the system has the knowledge of user's contacts, it has better chance to correctly tag the person names in communication (e.g. calling, texting, email) scenarios. Assuming that "May Lee" is a name in the user's contact list, when the user says call May, NLU should be able to tag "May as *person_name*, as oppose to *time* or *business_name* or assigning no tag at all. In another example, if the system knows Netflix is an application installed on the user's device, when the user says "go to netflix, the system should open Netflix application instead of navigating to netflix.com. Note that "Netflix" could be tagged as either a *website* or an *application_ name* and the corresponding system actions and the user experience would be different. Without the user specific data, the system may not perform as accurately as it should. In both cases, NLU models may not correctly tag person names and application names, which can be very diverse, including names from other language, ambiguous app names or novel names created for a new app or website.

It is almost impossible for NLU to have a large coverage of all possible person names and application names. User specific data is critical for NLU components to better understand the user's intent. Even though it has been evident that personalizing the NLU systems should lead to improved system quality, there is increased concern for sharing and protecting personal data.

In this paper, we propose a novel framework to ingest various personal information to personalize NLU models and keep personal data protected. NLU models are usually trained using domain specific queries with semantic annotation. In addition to the regular features, such as word n-grams, lexicons, etc. we make use of personalization features to train the NLU models in order to accurately predict user's intents. We focus on using personal contacts and application list on their devices for intent detection and slot tagging tasks. The same approach can be applied to other personal information. To the best of our knowledge, this is the first attempt to utilize the user data to personalize language understanding models in a very generic and scalable way.

The rest of this paper is organized as follows. In the next section, we discuss personalization related work in relevant NLP areas. In Section 3, we describe the approach to build personalized language understanding models. In Section 4, we provide experimental results when personal signals are used in

our NLU system. We conclude our work and discuss future personalization work in Section 5.

2. Related Work

Personalization has started to make its way into speech recognition. There were several studies that showed improvements in speech recognition accuracy through personalization. In [8, 9], language model personalization techniques are explored to improve voice command and dictation accuracy for speech recognition on mobile devices. User's personal contacts are integrated into language model. Language model personalization is achieved through a combination of vocabulary injection and on-the-fly language model biasing without significant adverse computational overhead [10, 11] . In [12], personalized word-phrase-entity language models are shown to achieve better performance compared to traditional class-based LMs for speech recognition.

Besides speech recognition, personalization has been investigated in other areas such as recommendation systems and web search. In [13], a personalized recommendation system and dialog system is proposed. Individual long-term user preferences are unobtrusively obtained in the course of normal recommendation dialogs and used to direct future conversations with the same user. In [14], researchers propose to use the user's long-term search history and location to effectively personalize auto-completion rankers for web search. Their results suggest that supervised rankers enhanced by personalization features can significantly outperform the popularity-based baselines. A collaborative personalized search is presented where a statistical user language model is trained to integrate the individual model, group user model and global user model together to enhance the performance of personalized search [15].

In language understanding, the most closely related work is [16]. Chen, et al., built personalized intent detection models using the user's behavioral history (a set of apps previously launched in the ongoing dialog) to estimate the probability distribution of all intended apps for each dialog turn. The personalized NLU is able to improve intent prediction using inferred app preference, e.g. a user may prefer "Outlook" over "Gmail", based on which app the user most often sends email after taking pictures. In contrast, we propose a generic method to use personal information to improve language understanding models in classification and tagging tasks. The NLU models are not replicated for each user. Instead, the models are shared by all the users, however at runtime feature extraction is personalized by using user specific personal data (i.e. contacts and application names). The models behave effectively in a personalized manner when detecting intents and tagging slots.

3. Personalized Natural Language Understanding

In this study, we focus on intent classification and slot tagging tasks in NLU. In addition to features used in baseline models, we introduce an approach to injecting personal information as external and dynamic features into NLU modeling. For each supported domain, we collect a set of in-domain queries. Queries are then annotated manually using a predefined semantic schema, which defines the semantic space represented in the query. The semantic representation of a user query is a triple of domain, intent, and slot list.

$$\langle \textit{domain, intent, slot\_list} \rangle$$

A slot list is a list of key-value pairs:

$$\langle \textit{slot\_type, slot\_value} \rangle$$

Each query belongs to one or more domains, such as *communication, device-control, music*, etc. Also, a query is classified into one of the pre-defined domain-specific intents, such as *send_email, make_call, open_application*, etc. Semantic slots associated with each intent, specify detailed parameters needed for back-end service call to fetch the relevant content or complete a task.

Typically, a machine learned classifier such as multi-class support vector machine (SVM) or deep learning models are built for both domain and intent classifiers. Domain classifiers are used to identify the domains a query belongs to, and for each domain, a domain-specific intent model is trained to distinguish the intents within the domain. Sequence taggers (e.g. CRF, RNN) are trained to label each word in a query with its semantic type.

The architecture of personalized NLU is illustrated in Figure 1. There are two separate processes executed offline. First, the user's personal information (such as user contacts and application lists) is encrypted and uploaded to the personalization server. The personalization server is used to securely store personal information. Secondly, models are trained offline using annotated training data, feature configuration (what features to use and how to extract them) and a mock user profile which contains a list of artificial person names and application names.

At run time, the NLU server calls the personalization server with a natural language (NL) query, together with a server token (an identifier of the NLU Server) and the user's token. The personalization server finds personal information related to the current user and returns personalization features matched in the query. For example, potential person names, application names appear in the query. The NLU Server uses the personalization features and features extracted at query level and session level to decode the query.

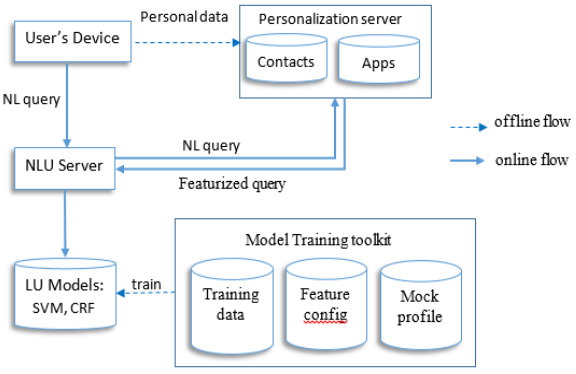

Figure 1: Personalized NLU architecture.

3.1. Features

Different types of features are used for intent classification and slot tagging to improve model robustness, accuracy and generalization, such as word n-gram, skip-gram features. Lexicons are used to group entities belonging to the same class to improve model's coverage [17]. Word class and embedding features are also adopted to improve the model's accuracy [18, 19]. Character n-grams are included to handle incomplete text queries [19].

Table 1: *Features used in baseline and personalized models.*

Features	Baseline models	Personalized models
word n-grams	✓	✓
lexicons	✓	✓
word classes	✓	✓
letter n-grams	✓	✓
contextual features	✓	✓
contact matched	✗	✓
application matched	✗	✓

Contextual features are used to track semantic frames and dialog states from previous turns [5].

We introduce a new feature set to capture user's personal information. This feature set includes data such as user's contacts, application list, play list, to-do list, browsing history, etc. NLU models are trained offline to weight each feature. Personalization features are extracted from the user profile at run time for decoding. The personalization feature function checks if a sequence of words in the query matches any entry in the target feature set. If matched, feature value 1 is assigned to all matched words; otherwise 0 is assigned. For example, the last three words in the query "open visual studio 2015" all have feature value 1 if the user has installed the application. In another sample query"call xiaohu liu", the personal contact feature function sets feature value 1 to last two words if there exists a contact named "xiaohu liu" in the user's address book. As user profiles vary, the feature value is user dependent. Hence the same query can be decoded differently.

The features used for both baseline models and personalized models are listed in Table 1.

3.2. Intent classification

In this study, we use support vector machine (SVM) as intent models. A separate multi-class intent model is built for each domain. The intent model for the *communication* domain classifies queries into intents such as *make_call, send_email, send_text*, etc. The intent model for the *device-control* domain covers *open_app, open_setting, change_setting, check_wifi*, etc., which controls the device functionality (without needing service data).

The SVM models are trained using queries annotated with domain specific intents. Intent models are personalized by adding personalization features. Intuitively, a query that contains a person name matching a contact name in the users contact list should have higher probability of *communication* intents, and a query with an application name matching an app installed on the device is more likely to be an *open_application* intent.

3.3. Slot tagging

Slot taggers are trained to tag all words of a query with slot types, which are predefined by the domain specific schema. For example, the query *"text Ashley that I am home"* is tagged as *"O person_name O message message message"*, where "O" is used to label a word outside the schema. A conditional random field (CRF) model is built as a slot tagger in each domain. Slot tagging and intent classification share the same feature set.

Table 2: *Intents and slots in test domains.*

Domains	Sample intents	Sample slots
communication 31 intents 21 slots	find_contact add_contact make_call send_email send_text	person_name email_address relationship message phone_no
device-control 26 intents 5 slots	open_application close_application locate_device open_setting power_off	app_name device_type setting_type position_ref media_type

3.4. Model training

Ideally one would need to have labeled data sets tied to each user's account containing personal data during model training. However, it is challenging to train personalized NLU models in this manner, since user's personal data is not accessible. The annotated data is not tied to specific users either. We create a mock user account with artificial personal content including application names and person names extracted from annotated training and test data. However, we do not add all entities from training data to the mock user profile so that the personalization features are not heavily weighted over other existing features. Our experiments over a development set show that we can achieve the best performance when the mock user profile only contains 60% of application names and person names in training data. With the mock account, we do not need any real user profile to train personalized models.

The mock user profile is only used to train personalization features but not used at run time. At run time, the feature values are determined by the personalization server based on the query and the real user's profile.

3.5. Decoding

The NLU server first sends the query to the personalization server to extract personalization features. Only matched features are sent back to the NLU server to minimize the message payload and avoid leaking private information. The NLU server runs local query and session level feature extraction in parallel for all other features, i.e. n-grams, lexicons, etc. Once all features are computed and concatenated, intent classification and slot tagging are executed simultaneously.

4. Experiments

4.1. Experimental setup

We build personalized models in two target domains for evaluation: *communication* and *device-control*. The first domain covers communication scenarios, such as *make_calls, send_texts, send_email, find_contacts*, etc. The second domain contains user's intents to control device functionality such as *open_application, close_application, change_brightness, change_settings, check_wifi_settings*, etc. The intent and slot distribution in two domains are shown in Table 2.

The data sets are presented in Table 3. Test set A is randomly sampled from Cortana user logs and annotated for testing. Test set B is derived from the test set A by replacing target slots, such as *application_name* and *person_name* with values from the mock user's profile, but not covered in training set.

Table 3: *Data for training and testing.*

Domains	Training set	Test set A	Test set B
communication	158K	11.6K	11.6K
device-control	66K	6K	6K

Table 4: *Offline evaluation results of communication domain.*

Models in the communication domain	Test set A	Test set B
Baseline intent model accuracy(%)	84.8	83.9
Personalized intent model accuracy(%)	84.8	85.0
Baseline slot model F1 score	96.0	95.5
Personalized slot model F1 score	96.4	97.3

This is the target scenario where we try to verify our personalized models by testing on data with slot values not seen at training time. Models have not seen these words/phrases in the training set, but they are covered by the user's profile data. We expect personalized models will perform better in the target test set B.

Both baseline models and personalized models use the same training data set. Only personalized models are trained with additional features extracted from the mock user profile.

4.2. Evaluation and discussion

We evaluate baseline models and personalized models in *communication* and *device-control* domains. Models are compared for both intent classification and slot tagging tasks. The experimental results of the *communication* domain are shown in Table 4 and results in the *device-control* domain are presented in Table 5. Intent classification is measured using accuracy metric and slot tagging is measured using F1 score.

In the *communication* domain, compared to the baseline model, the slot accuracy F1 score increase from 96 to 96.4 when tested on test set A. The personalized intent model has the same performance as the baseline model on test set A. The personalized models show much better results when tested on test set B, which is the target scenario we want to focus on. In test set B, the person names are not covered by training data, which is a more realistic scenario. Because the training data size is quite large and training feature set is fairly rich, the baseline models perform relatively well. However, personalized models can still outperform the baseline with significant F1 score gain.

The performance gains of personalized models in the *device-control* domain are much larger on both test sets. The performance is especially low on test set B, where application names are all new to the baseline model. With personal features (application names), personalized models achieves significant gains. One of the reason for large gains is that application names are more diverse than person names in any given locale. Also queries with application names usually do not follow limited syntactic patterns observed in the *communication* domain (e.g. call *person_name*, send an email to *person_name*).

In addition to the offline tests above, we also evaluate the personalization method with online queries. We first randomly select 2000 queries logged in the *communication* domain with *person_name* tagged by online personalized LU models. Then we run baseline models against the query set offline. The results are manually judged to compare performance. We observe that personalized intent and slot models outperform baseline as shown in Table 6.

Table 5: *Offline evaluation results of device-control domain.*

Models in the device-control domain	Test set A	Test set B
Baseline intent model accuracy(%)	98.9	88.8
Personalized intent model accuracy(%)	99.3	97.0
Baseline slot model F1 score	89.3	79.1
Personalized slot model F1 score	94.9	96.5

Table 6: *Online evaluation of communication domain.*

Models in communication domain	Scores
Baseline intent model accuracy(%)	87.5
Personalized intent model accuracy(%)	88.6
Baseline slot model F1 score	96.0
Personalized slot model F1 score	97.8

We analyzed some of the error patterns in the communication domain. The italic words below are tagged as *person_name* by baseline models, but they are corrected by personalized models. The baseline model tends to tag the two words following "contact, email, text, call", as *person_name*.

- contact *lenses*
- email johnson *ready*
- call tony ava *nokia*
- respond to rob *will* be five minutes late
- text abinash *backpack* on top of your car

Similarly, in the *device-control* domain, the baseline model tends to tag words after "open, switch, start, launch" as application names. The words in italic below are not application names, but tagged as such by the baseline model.

- switch *zoo*
- start *the website*
- launch *trampoline park*

5. Conclusion and Future Work

In this study, we proposed a method to use personal information to improve a natural language understanding system and keep personal data secure and protected. We used personalization features created from user's profile for both intent classification and slot tagging tasks. In our experiments with personal contacts and personal application lists as the personal data sources, we achieved significant performance gains on both tasks. As part of our future work, we are planning to add calendar entries, play lists, browse history, etc. as additional personal information sources to improve the performance of natural language understanding systems. We will also investigate using personalization information in dialog management. For example, in response to a query (e.g. "open Netflix"), the dialog manager can open the application, if it is installed on user's device (e.g. smart phones); if not, it can take the user to the sign-in page and the website of that application.

6. Acknowledgments

The authors would like to thank our colleagues: Edward Guo, Jacky Kang, Xuecheng Zhang, and Zhaleh Feizollahi, for helpful research discussions and support for the implementation.

7. References

[1] Ruhi Sarikaya, "The technology powering personal digital assistants", *Interspeech Keynote: http://www.superlectures.com/interspeech2015/the-technology-powering-personal-digital-assistants*, Dresden, Germany, 2015.

[2] S. Young, M. Gasic, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, K. Yu, "The hidden information state model: a practical framework for POMDP-based spoken dialogue management," *Computer Speech & Language*, 2009, 24 (2), pp.150-174.

[3] J.P. Robichaud, P. Crook, P. Xu, O. Z. Khan, R. Sarikaya, "Hypotheses ranking for robust domain classification and tracking in dialogue systems," *In Proc. of Interpseech*, Singapore, September 2014.

[4] O. Z. Khan, J.P. Robichaud, P. Crook, R. Sarikaya, Hypotheses Ranking and State Tracking for a Multi-Domain Dialog System using ASR Results," *Interspeech*, Dresden Germany, September, 2015.

[5] Puyang Xu and Ruhi Sarikaya, "Contextual domain classification in spoken language understanding systems using recurrent neural network", *Proceedings of ICASSP*, Florence, Italy, 2014.

[6] Gokhan Tur and Renato De Mori, "Spoken language understanding: systems for extracting semantic information from speech", John Wiley and Sons, 2011.

[7] A. Deoras, R. Sarikaya, "Deep Belief Network based Semantic Taggers for Spoken Language Understanding", *In Proc. Interspeech*, Lyon, France, 2013.

[8] Ian McGraw, Rohit Prabhavalkar, Raziel Alvarez, Montse Gonzalez Arenas, et al., "Personalized speech recognition on mobile devices", *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[9] Andrew Fowler, Kurt Partridge, Ciprian Chelba, Xiaojun Bi, Tom Ouyang, Shumin Zhai, "Effects of language modeling and its personalization on touchscreen typing performance", *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2015.

[10] Petar Aleksic, Cyril Allauzen, David Elson, Aleksandar Kracun,Diego Melendo Casado,and PedroJ. Moreno, "Improved recognition of contact names invoice commands," *ICASSP*, 2015

[11] Keith Hall, Eunjoon Cho, Cyril Allauzen, Francoise Beaufays, Noah Coccaro, Kaisuke Nakajima, Michael Riley, Brian Roark, David Rybach, and Linda Zhang, "Composition-based on-the-y rescoring for salient n-gram biasing," *INTERSPEECH*, 2015

[12] M. Levit, A. Stolcke, R. Subba, S. Parthasarathy, S. Chang, S. Xie, T. Anastasakos, and B. Dumoulin, "Personalization of word-phrase-entity language models," *Proceedings of Interspeech*, 2015.

[13] Cynthia A. Thompson, Mehmet H. Gker, Pat Langley, "A personalized system for conversational recommendations", *Journal of Artificial Intelligence Research*, Volume 21, 2004.

[14] Milad Shokouhi, "Learning to personalize query autocompletion", *Proceedings of the 36th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2013.

[15] Gui-Rong Xue, Jie Han, Yong Yu, and Qiang Yang, "User language model for collaborative personalized search", *ACM transactions on information systems*, Volume 27, 2009.

[16] Yun-Nung Chen, Ming Sun, Anatole Rudnicky, Alexander I. Gershman, "Leveraging behavioral patterns of mobile applications for personalized spoken language understanding", *Proceedings of The 17th ACM International Conference on Multimodal Interaction,* 2015.

[17] Xiaohu Liu, Ruhi Sarikaya, "A discriminative model based entity dictionary weighting approach for spoken language understanding", *Spoken Language Technology Workshop (SLT)*, 2014.

[18] Ruhi Sarikaya, Asli Celikyilmaz, Anoop Deoras, and Minwoo Jeong, "Shrinkage based features for slot tagging with conditional random fields", *Proceedings of Interspeech*, Singapore, 2014.

[19] Xiaohu Liu, Asli Celikyilmaz, Ruhi Sarikaya, "Natural language understanding for partial queries", *Proceedings of Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2015.