Personalizing Recurrent Neural Network Based Language Model by Social Network

Article *in* IEEE/ACM Transactions on Audio, Speech, and Language Processing · December 2016

DOI: 10.1109/TASLP.2016.2635445

Bo-Hsiang Tseng
National Taiwan University
5 PUBLICATIONS **3** CITATIONS

Yu Tsao
Academia Sinica
129 PUBLICATIONS **500** CITATIONS

Personalizing Recurrent Neural Network Based Language Model by Social Network

Hung-yi Lee, Bo-Hsiang Tseng, Tsung-Hsien Wen and Yu Tsao

***Abstract*—With the popularity of mobile devices, personalized speech recognizers have become more attainable and are highly attractive. Since each mobile device is used primarily by a single user, it is possible to have a personalized recognizer that well matches the characteristics of the individual user. Although acoustic model personalization has been investigated for decades, much less work has been reported on personalizing language models, presumably because of the difficulties in collecting sufficient personalized corpora. In this paper, we propose a general framework for personalizing recurrent neural network based language models (RNNLMs) using data collected from social networks, including the posts of many individual users and friend relationships among the users. Two major directions for this are model-based and feature-based RNNLM personalization. In model-based RNNLM personalization, the RNNLM parameters are fine-tuned to an individual user's wording patterns by incorporating social texts posted by the target user and his or her friends. For the feature-based approach, the RNNLM model parameters are fixed across users, but the RNNLM input features are instead augmented with personalized information. Both approaches not only drastically reduce the model perplexity, but also moderately reduce word error rates in n-best rescoring tests.**

***Index Terms*—Recurrent Neural Network, Personalized Language Modeling, Social Network**

I. Introduction

A current major trend is the personalization of applications and services for each individual user. Today, when users enter search terms into the search engine, users receive different results based on their specific interests and backgrounds [1]–[4]. Shopping sites recommend to consumers the products they are likely to be interested in [5]–[7]. Language learning systems provide learners with tailored learning materials to make language learning more effective and attractive [8], [9].

Personalized language models (LMs) are useful in several applications [10]–[12]. The most direct application for personalized LMs is personalized speech recognition. Because utterances produced by different users have different acoustic and linguistic characteristics, acoustic models (AMs) and LMs trained to work reasonably well for a large group of users may not perform as well for an individual. Therefore, personalized models that are well-matched to the characteristics of individual users can yield much better recognition performance. The concept of personalized speech recognition is highly attractive now due to the popularity of mobile devices such as smart phones and wearable clients. Because each mobile device is used primarily by a single user, the user could experience much better recognition performance if the device were equipped with a set of personalized AMs and LMs for speech recognition. In addition to speech recognition, personalized LMs have other applications. For instance, if the authorship of a document is in doubt, a personalized LM can serve as a proxy of one's writing style to identify the authorship of the document [10].

The aim of LM personalization here is different from that of LM adaptation which has been studied for decades [13]–[20]. Whereas LM adaptation focuses primarily on the problem of cross-domain or cross-genre linguistic mismatch, LM personalization as described here focuses on cross-individual linguistic mismatch. One reason that LM personalization has not been widely studied before is likely because it requires corpora produced by many different individuals. In earlier years, because of the lack of corpora suitable for the task, LM personalization was not easily realized.

Social networks on the Internet have been very popular among people of all groups for sharing information, ideas, interests and experiences, as well as interacting with each other in different ways. As social media blossoms today, and given the fact that each user is a part of the social network, we can take advantage of the huge quantities of texts left on the network by large numbers of users with known relationships. Because the text messages posted by users on social networks are good sources of individual wording habits, linguistic patterns, interests, and so on, these data can be used to estimate personalized LMs for social network users [21]. This leads to the fact that personalization, which has been intensively studied in areas such as retrieval and language learning, is now feasible for language modeling as well.

However, typically a large amount of text data is required to train a high quality LM. Therefore, exploiting social network data may be helpful only for very active users with large amounts of text posts on social networks. For general users who post only limited numbers of text messages on social networks, there is likely too little information to train a high quality personalized LM. Fortunately, the relationships among users available in social networks can also be used in LM personalization. Because it is reasonable to assume that users with close relationships share common subject topics, wording habits and sentence patterns, for users with small amounts of text in the social network, it can be helpful to use the user's friends' data also when building the personalized LM.

In this paper, we propose a general framework for personalizing recurrent neural network-based language models (RNNLMs) using text posted by many individual users and

This work was supported in part by Ministry of Science and Technology (MOST), R.O.C., under Contract 104-2218-E-002-021-MY2. We would also like to acknowledge the help of our colleagues, Ching-feng Yeh and Jeremy Chiaming Yang, for their help with the deep leanring based speech recognition.

their friend relationships collected from social networks. There are two major directions for RNNLM personalization: the model-based approach [22] and the feature-based approach [23][1]. In model-based RNNLM personalization, RNNLM parameters are fine-tuned toward an individual user's wording patterns by incorporating social texts posted by the target user and his or her friends. Therefore, in this approach each user has a set of personalized model parameters. For the feature-based approach, the RNNLM parameters are fixed across users but the input features are augmented with personalized information. Both RNNLM personalization approaches not only drastically reduce the model perplexity in preliminary experiments, but also moderately reduce the word error rates in n-best rescoring tests.

The rest of this paper is structured as follows. Language model personalization is defined in Section II, and N-gram-based LM personalization is reviewed in Section III. The RNNLM is introduced in Section IV, and model- and feature-based RNNLM personalization are respectively presented in Sections V and VI. The experimental setup and results are in Sections VII, VIII and IX, and we conclude in Section X.

II. Language Model Personalization

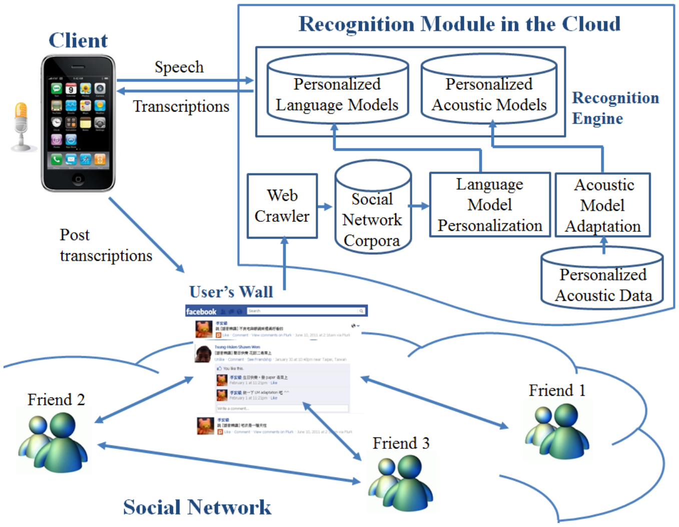

Client
Speech
Transcriptions
Recognition Module in the Cloud
Personalized Language Models
Personalized Acoustic Models
Recognition Engine
Web Crawler
Social Network Corpora
Language Model Personalization
Acoustic Model Adaptation
Personalized Acoustic Data
Post transcriptions
User's Wall
facebook
Friend 2
Friend 1
Friend 3
Social Network

Fig. 1: *Application scenario of LM personalization as considered in this paper. Mobile application used for users to post messages over social networks using speech. Includes cloud module with recognition engine, as well as web crawler to collect text from social network for LM personalization.*

With the mass production and rapid proliferation of smartphones in recent years, voice access has become desirable for many applications [24]. It is desirable to allow the user to enter his or her input directly by voice, given that the smartphone itself is a voice-operated device. Shown in Fig. 1 is the application scenario for LM personalization as considered in this paper: a mobile application for users to post messages over social networks by speech. We provide over the cloud a speech recognition module with a recognition engine. Smartphone users utilize the speech recognition service to post text to the social network using voice. For the problem considered here, however, since different users tend to post messages about many relatively disjoint topics on the social networks with significantly different N-gram statistics, LMs trained to work reasonably well for large groups of users may not perform as well for individual users. In addition, what users tend to post on social networks tends to be casual, with non-standard grammar, which further complicates recognition. However, since most text on social networks is relatively casual, users may have a slightly higher tolerance for recognition errors when using the application; still, it is always helpful to mitigate recognition errors. Therefore we adopt personalized speech recognizers, which are very helpful because smartphones are typically used by a single user. For each user the personalized recognition module maintains a personalized LM and AM. The recognition module transcribes the utterances produced by the user and returns the transcriptions. When the transcriptions are shown on the screen, the user decides whether to post it on the *Wall*[2] or not. If necessary, users manually correct inaccurate transcriptions.

For better accuracy from the personalized recognizer, the user registers for personalized speech recognition from our voice access service via his or her Facebook account and grants our application the authority to collect acoustic and linguistic data for personalization. The acoustic utterances produced by each individual user are collected for AM personalization, more commonly referred to as AM adaptation. AM adaptation [25]–[27] has been investigated for decades and yields impressive improvements with many approaches based on either HMM/GMM or CD-DNN-HMM [28]; however this is out of the scope of this paper.

In this paper, we focus on personalizing LMs, primarily RNNLMs. A crowdsourcing mechanism is designed to collect text and other information on social networks from users for LM personalization. Crowdsourcing [29], [30] has been applied for several purposes in many different fields. For example, a crowdsourcing approach [31] was proposed to collect queries in an information retrieval system considering temporal information. The MIT movie browser [32], [33] used Amazon's Mechanical Turk, the most well-known crowdsourcing platform, to build a crowd-supervised spoken language system. In our case, an implicit crowdsourcing [30] is at play. A web crawler is implemented in the recognition module over the cloud for collecting text corpora[3] from the social network to train the personalized LM of each user. The crawler collects from the social network the following text corpora. For each user u, the texts of his or her social network posts form the *personal corpus* $\mathcal{S}_u$[4] for the user. In addition, the posts of user u's friends in the social network are collected to form corpus $\mathcal{F}_u$, below referred to as the *friends corpus*. The collection of all the text posts collected from the Internet, including posts authored by many different users, forms the *background*

[1] Although the RNNLM personalization approaches were proposed in previous work [22], [23], in this article we provide more detailed experimental results and analysis.

[2] The *Wall* is a place to post one's messages.

[3] Because of privacy issues, only information granted by the user is accessible to the crawler.

[4] $\mathcal{S}$ stands for *self*.

corpus $\mathcal{B}$. The user's personalized LM is thus based on corpora $\mathcal{S}_u$, $\mathcal{F}_u$, and $\mathcal{B}$.

III. Personalization of N-gram-based Language Models

The N-gram-based LM is the most common language modeling approach [16], [34], [35]. Given a word sequence of length T, $\{w_t : 1 \leq t \leq T\}$, the N-gram-based LM represents the probability of the word sequence as

$$P(w_1, ..., w_T) = \prod_{t=1}^{T} P(w_t|h_t), \tag{1}$$

where h_t is the history of the word w_t, and $P(w_t|h_t)$ is the probability of observing word w_t in a word sequence given the history h_t.

In personalization, each user u has an N-gram-based LM fitted to the statistics of the text he or she posts on the social networks. Because the personal corpus $\mathcal{S}_u$ collected by the web crawler in Fig. 1 is the best example of what this user will say in the social network in the future, the most direct way to achieve personalization is to directly estimate the N-gram probabilities $P_u(w_t|h_t)$ from each $\mathcal{S}_u$, and use these as the N-gram probabilities of the user's personalized LM. However, because $\mathcal{S}_u$ is usually small, data sparsity renders the personal LM thus learned unreliable.

To address the data sparsity issue, we borrow the framework of LM adaptation. The basic framework for LM adaptation considers two text corpora: the first is the adaptation corpus, which is "in-domain" or up-to-date with respect to the target recognition task, but generally too small to train a robust standalone LM; the other is a large background corpus, which is generally not sufficiently related to the target task or is perhaps outdated. In traditional N-gram models, the two corpora are interpolated; various methods are used to estimate their interpolation weights [36]–[39]. In the LM personalization described here, the large background corpus $\mathcal{B}$ containing the posts of a large number of users is available and is used to train a robust LM, but the LM is not related to any single user in particular. The personal corpus $\mathcal{S}_u$ plays the role of the adaptation corpus in LM adaptation to interpolate with the LM trained from the background corpus $\mathcal{B}$ as

$$P'_u(w_t|h_t) = \alpha_u P_u(w_t|h_t) + (1 - \alpha_u) P_b(w_t|h_t) \tag{2}$$

where $P_u(w_t|h_t)$ are the N-gram probabilities estimated from personal corpus $\mathcal{S}_u$, $P_b(w_t|h_t)$ are the N-gram probabilities estimated from background corpus $\mathcal{B}$, and $P'_u(w_t|h_t)$ are the N-gram probabilities of the new personalized LM obtained by the linear interpolation of $P_u(w_t|h_t)$ and $P_b(w_t|h_t)$. α_u in (2), the interpolation weight of the two N-gram probabilities, is user-dependent with its value determined using the validation set of each user.

The posts of u's friends $\mathcal{F}_u$, which includes content related to what u has posted, can be used in LM personalization as

$$P''_u(w_t|h_t) = \beta_u P'_u(w_t|h_t) + (1 - \beta_u) P_f(w_t|h_t), \tag{3}$$

where $P'_u(w_t|h_t)$ is in (2), $P_f(w_t|h_t)$ are the N-gram probabilities estimated from $\mathcal{F}_u$, and $P''_u(w_t|h_t)$ is the interpolation of $P'_u(w_t|h_t)$ and $P_f(w_t|h_t)$. Interpolation weight β_u in (3) is also a user-dependent parameter tuned using validation sets. $P''_u(w_t|h_t)$ serves as the N-gram probabilities of another personalized LM in which not only the posts of user $\mathcal{S}_u$ are represented but also the posts of his of her friends $\mathcal{F}_u$.

IV. Recurrent Neural Network based Language Model (RNNLM)

The lack of a natural strategy to model long-range dependencies [40]–[42] and the potentially error-prone back-off behavior [43] limit the performance of N-gram-based LMs. Therefore, researchers have long sought a language modeling approach to replace N-grams. Recently, several studies have shown that neural-network-based LMs (NNLMs) [44], [45], [45]–[47] improve on N-gram-based LMs by taking advantage of the learned distributed representations of word histories. Among NNLMs, the recurrent neural network based LM (RNNLM) [48]–[53] has especially drawn attention in its ability to memorize arbitrary length of histories in a recurrent structure and thus elegantly model long-range dependencies.

The structure of RNNLM is shown on the right half of Fig. 2. The basic RNNLM comprises three layers: the input layer, the hidden layer, and the output layer. The input layer represents the t-th word in a sentence, w_t, using a 1-of-N encoding. The context vector s_t is the distributed representation of the history word sequence, with a recurrent connection considering the time-delayed context vector s_{t-1}. The output layer y_t then generates the probability distribution of the next word. In order to provide complementary information such as part-of-speech tags, topic information, or morphological information to the input vector w_t, a context-dependent RNNLM variant [50] adds an additional feature layer f_t to the network and connects it to both the hidden and output layers. Therefore, the network weights to be learned are the matrices $\mathcal{W}$, $\mathcal{S}$, $\mathcal{O}$, $\mathcal{H}$, and $\mathcal{G}$. The learning process maximizes the likelihood of the training data using the back-propagation through time (BPTT) algorithm. Typically, a validation set is used to control the training epochs and learning rates.

For our application scenario, we adopt RNNLMs for the following reasons.

1) The number of social posts we obtain for each individual are still too sparse for robust use with N-gram LM adaptation approaches. However, for RNNLM, the distributed representation of word histories mitigates the sparsity problem. Because RNNLM projects the originally disjoint history word sequences onto the same continuous space, even though a history word sequence has not been seen before, by projecting it onto the correct point in the space, the probability distribution of the next word can still be accurately estimated.
2) Online messages or social posts tend to be relatively casual, and thus, do not usually obey traditional grammar rules strongly. As a consequence, short-term dependencies may no longer be evident enough to predict the next word. The recurrent structure may help mitigate this issue as well, since it memorizes longer dependencies than N-gram LMs.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2016.2635445, IEEE/ACM Transactions on Audio, Speech, and Language Processing

3) The input of a RNNLM is a feature vector. As mentioned in previous work [50], [54], adding additional auxiliary features to augment the input 1-of-N encoding features is relatively easy and helpful. Therefore, in contrast to the N-gram-based LM, with an RNNLM is possible to apply feature-based personalization, which we describe in Section VI. The feature-based approach is shown to be better than the model-based approach in some aspects.

As a result, the combination of LM personalization and RNNLMs has strong potential. The personalization of RNNLM has two major directions: the model-based approach and the feature-based approach, described respectively in Sections V and VI.

V. Model-based RNNLM Personalization

For the data sparsity problem mentioned in Section III, we note that it is impossible to train a standalone RNNLM for each user given only the small amount of training data contained in the collected personal corpora $\mathcal{S}_u$. As a result, in model-based RNNLM personalization, we still resort to LM adaptation as in Section III. Here we first train a RNN-based background LM from background corpus $\mathcal{B}$, and then for each user u fine-tune the network parameters using the personal corpus $\mathcal{S}_u$ and friends corpus $\mathcal{F}_u$, resulting in a personalized RNNLM model for each user. Although the amount of adaptation data $\mathcal{S}_u$ and $\mathcal{F}_u$ can be small, it is believed that the distributed representation of RNNLMs can amplify the training efficiency because in the continuous space one training sentence informs the model about a combinatorial number of other sentences.

Model-based RNNLM personalization involves the following three steps:

1) Train a general-domain background RNNLM using the background corpus $\mathcal{B}$, which is split into a training set and a validation set. The likelihood on the training set is maximized, and the validation set is used to control the number of training epochs. Obtained here is a set of model parameters $\mathcal{W}^0$, $\mathcal{S}^0$, and $\mathcal{O}^0$ which are completely user-independent[5].
2) Given the target user's personal corpus $\mathcal{S}_u$, split it into training set $\mathcal{T}_u$ and validation set $\mathcal{V}_u$. Copy one background RNNLM and use BPTT to fine-tune parameters $\mathcal{W}^0$, $\mathcal{S}^0$, and $\mathcal{O}^0$ by maximizing the likelihood of the training set T_u while controlling the number of epochs using the validation set V_u. Fine-tuning yields the personalized model parameters $\mathcal{W}'$, $\mathcal{S}'$, and $\mathcal{O}'$.
3) Given friends corpus $\mathcal{F}_u$, we treat $\mathcal{F}_u$ as a complete training set, and maximize its likelihood, but control the number of epochs by the validation set $\mathcal{V}_u$ from personal corpus $\mathcal{S}_u$. This yields another set of personalized model parameters $\mathcal{W}''$, $\mathcal{S}''$, and $\mathcal{O}''$. By using $\mathcal{V}_u$ as the validation set to point the adaptation in the right direction, we constrain the influence of the friends corpus in LM personalization to be data-driven. For example, if the friends corpus $\mathcal{F}_u$ is unrelated to the personal corpus, the $\mathcal{V}_u$ from the personal corpus causes the training to stop earlier than it would otherwise.

VI. Feature-based RNNLM Personalization

There are some shortcomings in the model-based approach:

1) Even with the help of the social networks, the text corpora collected for adapting a background LM towards a personalized LM can still be insufficient[6]. As a result, the personalized LM thus obtained easily overfits to the limited data, and therefore would be expected to yield relatively poor performance for some users.
2) To train and store a personalized LM for every user is time-consuming and memory-intensive, especially considering that the number of users using the application will only increase in the future.

Considering the above-mentioned defects in the model-based approach, in this paper we propose feature-based RNNLM personalization. For the feature-based approach, instead of building a personalized RNNLM for each user, a single universal RNNLM is used by all users with input features being personalized. As shown in Fig. 2, a corpus of posts from a large group of users serves as the training data for the universal RNNLM. This universal RNNLM comprises three layers – the input layer, the hidden layer, and the output layer – as a general RNNLM, except for the input layer, which is not only the 1-of-N encoding of the t-th word, w_t, but also includes the user characteristic feature f_u. This user characteristic feature is connected to both the hidden layer s_t and output layer y_t. This structure parallels the context-dependent RNNLM variant [50], except for the context feature in the input layer, which is replaced by the user characteristic feature f_u. In contrast to context-dependent RNNLM, in which each word is augmented with the same context feature, in our approach, for each given user, each word produced by the user is augmented with the user's feature f_u. The user characteristic feature f_u enables the model to take into account each specific user. The network weights to be learned are the matrices $\mathcal{W}, \mathcal{H}, \mathcal{S}, \mathcal{G}$, and $\mathcal{O}$ in the right part of the figure. The standard training method is used, except that now the same words produced by different users in the training set are augmented by different user characteristic features.

During testing, given a new user, his or her characteristic feature is extracted to augment the 1-of-N word encoding, with which the universal RNNLM is used. Because the same words produced by different users are augmented with different features, given the same history word sequence, the universal RNNLM predicts different distributions of the next word for different users. In this way, personalization is achieved even though all users share the same universal RNNLM. The concept of combing heterogeneous features for personalization can be considered as a multi-task learning, which has been proven to effectively improve the generalization of deep neural network (DNN) by forcing it to learn more than one related task at a time [55]. This concept of personalized input features is similar to the i-vectors used in deep neural network (DNN)

[5] In the feature-based approach, the parameters $\mathcal{H}$ and $\mathcal{G}$ would also be trained here, as they are related to the feature layer.

[6] Some users have few posts and few friends.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2016.2635445, IEEE/ACM Transactions on Audio, Speech, and Language Processing

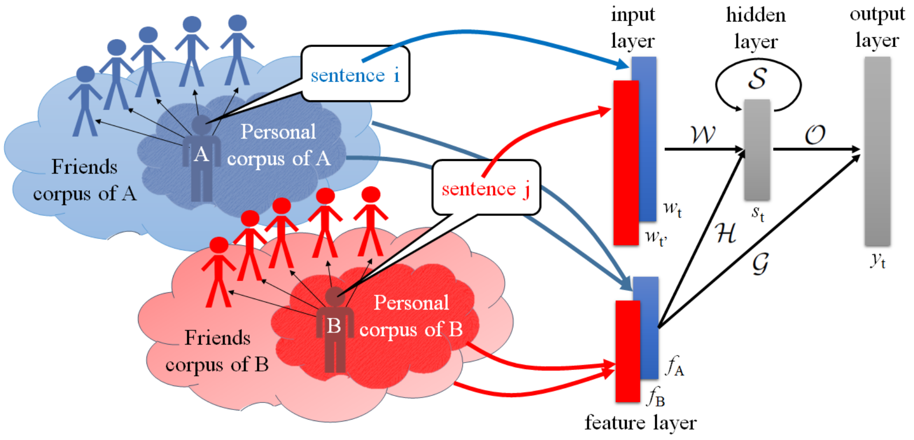


Fig. 2: *The architecture of the recurrent neural network LM (RNNLM) and feature-based personalization. When training with sentence i from user A, the user feature fed into the RNNLM is produced either as a topic distribution of the user's personal corpus or by searching over the user's personal/friends corpus for sentences with topic distributions similar to sentence i.*

based acoustic models [56], [57], in which the i-vector of each speaker is used to augment acoustic features such as MFCC features.

In the feature-based approach, the universal RNNLM trained from the social text produced by many users is less prone to overfitting because it is trained from a large training set obtained by aggregating the social texts of many users. Moreover, since the recognizer for each user requires only the user's characteristic features rather than an entirely new model, the new paradigm saves time during training and memory in real-world implementations.

Below we explore two approaches to extract the user characteristic features, described respectively in Sections VI-A and VI-B.

A. User-dependent Feature

In this approach, the personal corpus for each target user is viewed as a single document from which a topic modeling approach derives the topic distribution. The topic distribution of the personal corpus thus represents the language characteristics of the user and is considered the user characteristic feature f_u. That is, during training the universal RNNLM, the 1-of-N encoding of the words in a personal corpus are all augmented with the topic distribution of that personal corpus. Some social networks also include a significant amount of information such as user profiles, relationships, interactions, preferences, and interests. These are all valuable types of information to be considered for user-dependent features. However, since this information is not always available among all users in all social networks, here we do not take this information into account.

B. Sentence-dependent Feature

Considering the fact that the personalized corpus of a user can cover a wide variety of topics, and the user's topic may switch dynamically and freely in the personal corpus, the topic distribution for the whole personal corpus as a whole can differ significantly from that of each individual sentence within the personal corpus. On the other hand, even though the topic can switch freely within a user's personal corpus, we observe that typically at least a few sentences are needed before a specific topic is finished. Therefore, to form a feature that not only reflects the characteristics of the user but also a specific sentence, we exploit that part of the personal corpus whose topic distribution is similar to the sentence's. It is hoped that this will solve the problem of mismatch between the topic distribution of the whole personal corpus and each individual sentence.

With the above consideration, in the second approach, every sentence l in the personal corpus of a user u has its unique feature $f_{u,l}$ which is related not only to the user producing the sentence l but also to the sentence itself. Here the topic model is first used to infer the topic distribution of a sentence, and then we use this topic distribution to search over the personal corpus of the user to find an additional N sentences whose topic distributions are most similar to that of the sentence in question. Since this is limited to the personal corpus of the considered user, this search process is fast. While training the universal RNNLM, the average of the topic distributions of these N found sentences is taken as the user characteristic feature used to augment the 1-of-N encoding features. Therefore, the same words in different sentences of a personal corpus can have different user characteristic features. We can also extend the search space to include the user's friends corpora as well. As identifying the most similar sentences among the personal corpora of all users is very time consuming, it is not feasible here.

The major difference between the two approaches in Sections VI-A and VI-B lies in the concept of how to obtain a better language model. In the first approach, we assume the personal corpus of a user reflects his or her language characteristics; thus we use the whole personal corpus as the data from which to infer the topic distribution. In the second approach, we assume a user switches topics freely from sentence to sentence; thus we attempt to identify similar sentences to construct a feature that reflects language characteristics

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2016.2635445, IEEE/ACM Transactions on Audio, Speech, and Language Processing

not only for the user but for the sentence in question. This limits the data used to form the user characteristics to the N sentences found during the search process. During testing, the user characteristic feature is obtained in exactly the same way, except that the n-best list of an utterance is used with the topic model to generate the topic distribution for an utterance.

VII. Experimental Setup

A. Corpus & LMs

First of all, 0.5M sentences were collected from Plurk, a popular social networking site. The data from Plurk is the background corpus for training the N-gram-based LM, and the data was also used to train the topic model for use in extracting user characteristic features. Here we used the MALLET toolkit [58] to train a Latent Dirichlet Allocation (LDA) [59] topic model. The testing experiments were conducted on a crawled Facebook corpus. A total of 42 users logged in and authorized this project to collect their posts and basic information for research purposes. With their consent, all the data that can be accessed by the accounts of the 42 target users were collected including the posts of the target users, their friends and their friends' friends. This resulted in the personal data of 93,000 anonymous people and a total of 2.4M sentences. The number of sentences for each user among the 93,000 ranged from 1 to 8,566 with a mean of 25.7. The 0.5M sentences from Plurk plus the 2.4M sentences from Facebook (excluding the posts of the target users) formed the background corpus for training the non-personalized RNNLM. The larger background corpus was used to train the RNNLM because it usually needs more data to have good performance [49]. From the data crawled from the Facebook, the personal corpus $\mathcal{S}_u$ and friends corpus $\mathcal{F}_u$ for each target user u were obtained. In the data we collected, each target user was linked to an average of 250 other users.

The code-mixing phenomenon appears in the sentences collected from Plurk and Facebook. The sentences were produced in Chinese, but some words or phrases were naturally produced in English and embedded in the Chinese sentences. For example, in the sentence, *"兩天沒睡，paper 難產中 (I haven't been sleeping for two days. I am having difficulty writing the paper.)"*, the word "paper" was produced in English, while other parts of the sentence were in Chinese. For another example, *"小時候 C 沒學好，長大 debug 到天亮 (If you did not learn the C language well in your prime, you'll debug until dawn in your old age.)"*, "C" (referred to the C language) and "debug" were produced in English. The mix ratio for the Chinese characters and English words in the Facebook data is 10.5:1.

Both N-gram-based LMs and RNNLM models used the same lexicon. The lexicon we used consists of 18K English words and 46K Chinese words. The 46K Chinese words include all commonly used Chinese characters taken as mono-character Chinese words. All the Chinese words are composed of these mono-character Chinese words, so there is no OOV issue for Chinese here. The OOV rate for the English words will report below. The N-gram-based LMs were trained and adapted using the SRILM [60] toolkit, and the modified Kneser-Ney algorithm [61] was used for LM smoothing. RNNLM models were implemented with the RNNLM toolkit [62]. Perplexity (PPL) and word error rate (WER) are used to evaluate the models. In all the evaluations, the basic unit for alignment and calculation was character for Mandarin and word for English [63]. Because in Chinese different word sequences can correspond to the same character sequence, character is usually used as the basic unit instead of word.

B. Perplexity (PPL)

The posts of the 42 target users were used to evaluate the PPL of the proposed approaches. For each target user, 3/5 of his or her corpus was taken as the training set, 1/5 as the validation set, and the remaining 1/5 as testing data with which PPL was computed. There were a total of 12K sentences for testing. The OOV rate among the English words was 0.05%. The target users were divided into three groups for cross validation. The experimental results for PPL are shown in Section VIII.

C. N-best rescoring

In addition to PPL, we also evaluated the proposed approaches by rescoring the n-best lists of the speech recognition outputs. For the n-best rescoring experiments, we used 948 utterances of the 42 target users, from which we generated 1,000-best lists. There is no OOV words in this task[7]. To collect the utterances, we built a smart phone app. Through the app, the target users could see their recent posts (not involved in LM training). The target users were asked to read the posts, and the recorded utterances were sent to our server. The utterances collected in this way also have the code-mixing phenomenon as the text posts crawled from Facebook. This mimicked the scenario that the target users used speech to write their posts on the Facebook. By listening to the recorded utterances, we found that the utterances were recorded in a wide variety of environments, for example, office, street and so on. The speech recognition task considered here is very difficult because not only were the utterances recorded in various conditions with background noise, but some of the utterances were code-switched.

To transcribe the bilingual utterances, a bilingual phoneme set was used which is simply the combination of 37 Mandarin phonemes and 35 English phonemes. As a result, during decoding, an acoustic feature vector may belong to a state of an English phoneme or a state of a Mandarin phoneme. The lexicon will then constrain the possible phoneme sequences, and the language model will help in choosing the possible paths to form the n-best lists [64]. We need both Mandarin and English corpora to train the Mandarin and English phonemes. The Mandarin corpus we used was ASTMIC corpus [63] which contains read speech recorded under clean conditions produced by 95 males and 95 females, each reading 200 sentences, with a total length of 24.6 hours. For English

[7] This is reasonable because the English words embedded in the Chinese sentences in our task are usually common words.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2016.2635445, IEEE/ACM Transactions on Audio, Speech, and Language Processing

TABLE I: Perplexity (PPL) results. KN3 stands for Kneser-Ney trigram, while 'RNN/model' and 'RNN/feature' are for the model- and feature-based personalized RNNLMs, respectively. '$\mathcal{B}$', '$\mathcal{B}$+$\mathcal{S}$', and '$\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$' respectively indicate using only the background corpus $\mathcal{B}$, plus personal corpus $\mathcal{S}_u$, and plus friends corpus $\mathcal{F}_u$ in addition. 'UD' and 'SD' respectively indicate the extraction of user- and sentence-dependent features in Sections VI-A and VI-B. RNNLM results with hidden layer sizes of 50, 100, and 200 are listed ('h50', 'h100' and 'h200').

Perplexity		h50	h100	h200
(a)	(a-1) KN3, $\mathcal{B}$	343		
	(a-2) KN3, $\mathcal{B}$+$\mathcal{S}$	299		
	(a-3) KN3, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$	233		
(b)	RNN, $\mathcal{B}$	315	276	252
(c)	(c-1) RNN/model, $\mathcal{B}$+$\mathcal{S}$	271	247	230
	(c-2) RNN/model, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$	269	246	229
(d)	(d-1) RNN/feature, UD, $\mathcal{B}$+$\mathcal{S}$	313	290	270
	(d-2) RNN/feature, UD, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$	320	296	278
	(d-3) RNN/feature, SD, $\mathcal{B}$+$\mathcal{S}$	269	230	218
	(d-4) RNN/feature, SD, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$	229	215	211
	(d-5) RNN/feature, SD, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$, avg	192	159	165

corpus, we used the data collected for the Taiwan Asian English speech corpus project (TWNAESOP) [65], which is also a read speech corpus recorded under clean conditions produced by Taiwanese speakers, 229 males and 256 females, with a total length of 59.7 hours. Both the training corpora, ASTMIC and TWNAESOP, do not have the code-mixing phenomenon as the testing utterances. We have two sets of AMs, GMM-based and DNN-based:

- GMM-based: Tri-phone models were used, and each tri-phone model has three states with 24 Gaussians. The AMs were adapted using unsupervised MLLR speaker adaptation. 39-dimensional MFCCs were used as the acoustic features.
- DNN-based: We used DNN-HMM hybrid system. The DNN model has 4 hidden layers, and there are 2048 neurons for each layer. The output target for the DNN is 2500 senones. 69-dimensional filter bank features were used as the acoustic features, and the splicing windows were chosen as 4.

The LMs we used to generate the n-best lists were trigram models, adapted using the personal corpora as well as friends corpora with Kneser-Ney smoothing. The experimental results for WER are shown in Section IX.

VIII. Experimental Results: Perplexity (PPL)

A. *Perplexity (PPL) for Different Approaches*

Table I shows the perplexity (PPL) results. The personalized Kneser-Ney trigram LM results are in section (a). '$\mathcal{B}$', '$\mathcal{B}$+$\mathcal{S}$', and '$\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$' respectively indicate using only background corpus $\mathcal{B}$, plus personal corpus $\mathcal{S}_u$, and plus friends corpus $\mathcal{F}_u$ in addition. In row (a-1), only the background corpus $\mathcal{B}$ was used, so there was no personalization. Row (a-2) is the results of Eq. (2) in Section III, and row (a-3) is the results of Eq. (3). Clearly, personalization reduced PPL dramatically (rows (a-2) and (a-3) v.s. (a-1)); the posts of the target user's friends were helpful (rows (a-3) v.s. (a-2)) as well.

Row (b) and sections (c) and (d) are the results of RNNLM with hidden layer sizes of 50, 100, and 200 ('h50', 'h100', and 'h200'). Row (b) is RNNLM using only the background corpus ('$\mathcal{B}$') without any personalization. Personalized RNNLM based on the model- and feature-based approaches in Sections V and VI are respectively labeled 'RNN/model' in section (c) and 'RNN/feature' in section (d). In section (d), 'UD' and 'SD' respectively indicate the extraction of user- (Section VI-A) and sentence-dependent (Section VI-B) features in the proposed approach.

In row (b), larger hidden layer sizes yielded lower PPLs ('h200' < 'h100' < 'h50' in row (b)). For model-based personalization, we find that fine-tuning the model parameters with personal data was helpful (rows (c-1) v.s. (b)), and than fine-tuning the parameters with friend posts further lowered the PPL slightly (rows (c-2) v.s. (c-1)). For the user-dependent ('UD') feature-based approach in section (d), under the condition involving personal corpora ($\mathcal{B}$+$\mathcal{S}$), user-dependent feature-based personalization could not outperform the non-personalized RNNLM when the hidden layer sizes are 100 and 200 (rows (d-1) v.s. (b) for 'h100' and 'h200'). This confirms that the concern about the user-dependent feature-based approach in Section VI-B is reasonable. With the addition of the friends corpora ($\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$), the personal and friends corpora were assembled as a user with a large amount of data from which to extract user-dependent features. We find that with the user-dependent features, the addition of the friends corpora did not yield improved performance (rows (d-2) v.s. (d-1)). Because the friend posts covered a wide variety of topics, it is difficult to represent the friends with a single feature vector.

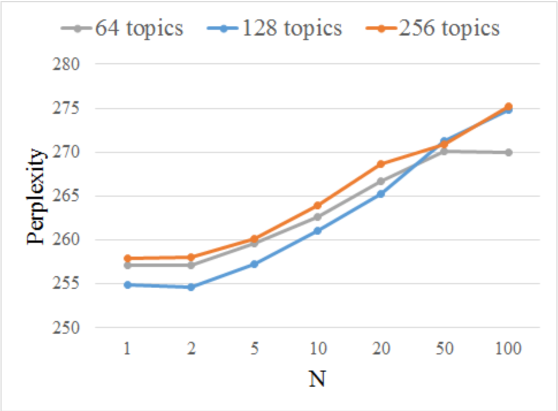


Fig. 3: *Perplexities for different numbers of LDA topics and different numbers of similar sentences (N) selected when building the user characteristic features in Section VI-B.*

For sentence-dependent ('SD') features, as mentioned in Section VI-B, only those N sentences (out of the user plus friends corpora) closest to the sentence under consideration were used to build the user characteristic features. Fig. 3 shows the PPLs for different N and different number of topics for

LDA[8]. The figure shows that there was almost no difference between $N = 1$ and $N = 2$, but as N increased beyond 2 the PPL also increased, suggesting a wide variety of topics even for the same user and his/her friends. We chose $N = 1$ for the following experiments.

In Table I, with the sentence-dependent ('SD') features, feature-based personalization consistently outperformed the model-based approach (rows (d-3), (d-4), (d-5) v.s. (c-1), (c-2)). This shows that it is more efficient to extract good features that characterize the user than it is to use personal data to learn a personalized RNNLM. We find that sentence-dependent features also outperformed the user-dependent features (rows (d-3), (d-4), (d-5) v.s. (d-1), (d-2)); this confirms our reasoning in Section VI-B about topic switching. With the addition of the friends corpora ($\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$), when extracting the sentence-dependent features, the search space extended over both the personal and friends corpora. In contrast to the user-dependent feature, the sentence-dependent feature is further improved with the addition of the friends corpora (rows (d-4) v.s. (d-3)), and it was also better than the model-based personalization when including the friends corpora (rows (d-4) v.s. (c-2)). When using the sentence-dependent feature we further averaged the user characteristic feature with the topic distribution of the sentence in question (RNN/feature, SD, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$, avg in (d-5)), yielding further PPL improvements (rows (d-5) v.s. (d-4)). With this best model obtained (RNN/feature, SD, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$, avg, 'h200'), the PPL was reduced by 34.5% in comparison to RNNLM without personalization (RNN, $\mathcal{B}$ in (b), 'h200') and by 27.9% in comparison to the model-based personalization approach with friends corpora (RNN/model, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$ in (c-2) in 'h200').

B. Effects of the user characteristic feature on RNNLM

To better understand the effects of the user characteristic feature on RNNLM, we offer the following analysis. First, we consider a real example from the Facebook data used in the experiments. In the Facebook data, user A left many posts about coffee, but user B never did so. This yielded very different user characteristic features for the two users. Here we used the user characteristic features mentioned in Section VI-B involving searching for the N closest sentences (that is, row (d-3) in Table I). Given the sentence *"一個長瓶的牛奶可以作三杯拿鐵 (A bottle of milk can make three cups of latte)"* produced by user A, we list in Table II the PPLs evaluated by a conventional RNNLM without the user characteristic feature and the personalized RNNLM with different user characteristic features. The RNNLM without personalization is in row (a). We find that the personalized RNNLM with the user characteristic feature of user A produced a drastically decreased perplexity (152 vs 322, row (b)) because of the well-matched characteristics, but when paired with the user characteristic feature of user B it yielded a significantly increased PPL (604 vs 322, row (c)).

Then we conducted a more general study. For each testing sentence, instead of using the personal corpus of the target user producing the sentence to build the user characteristic feature, a randomly picked target user was used (excluding the one producing the sentence). The results are shown in Table III. Rows (a) and (b) in Table III correspond to rows (b) and (d-3) for 'h200' in Table I respectively. Row (c) is the result using the personal corpus of a randomly picked user. We find that even with the randomly picked users, the PPL was still smaller (rows (c) v.s. (a)). This is because even a randomly picked user could have content similar to the sentence in question. However, using incorrect personal corpus still seriously degraded the performance due to the mismatch in user characteristics (rows (c) v.s. (b)).

TABLE II: Two users from the Facebook data. User A left many posts about coffee, but user B never did so. Given the sentence *"一個長瓶的牛奶可以作三杯拿鐵 (A bottle of milk can make three cups of latte)"* from user A's posts, this shows perplexities obtained without the user characteristic feature and with the user characteristic features of users A and B.

		Perplexity
(a) RNN, $\mathcal{B}$ (without user characteristic feature)		322
RNN/feature, SD,$\mathcal{B}$+$\mathcal{S}$	(b) feature of user A	152
	(c) feature of user B	604

TABLE III: Using the personal corpus of a randomly picked user to form the user characteristic feature. Rows (a) and (b) correspond to rows (b) and (d-3) for 'h200' in Table I respectively. Row (c) is the result from the randomly picked users.

		Perplexity
(a) RNN, $\mathcal{B}$ (without user characteristic feature)		252
RNN/feature, SD,$\mathcal{B}$+$\mathcal{S}$	(b) feature of target user	218
	(c) feature of random user	244

IX. Experimental Results: Word error rate (WER)

Below, the results for different personalization approaches with the GMM-based AMs are in Section IX-A, while the results for the DNN-based AMs are in Section IX-B.

A. GMM-based Acoustic Models

1) WER for different personalization approaches: Table IV reports the word error rates (WER) with the same notation as in Table I. The GMM-based AMs were used here. Due to the difficult speech recognition task considered here, the WER reported is relatively high, but the experiments here still demonstrate that personalizing LMs with the personal data from social network is helpful. Section (a) is for the three different trigram LMs without and with personalization. As expected, the trigram LMs yielded lower error rates given more adaptation data ((a-3)<(a-2)<(a-1)). We used the best adapted trigram LM (KN3, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$ in (a-3)) to generate 1000-best lists for RNNLM rescoring.

Section (b) shows rescoring results using RNNLM without personalization, and sections (c) and (d) are for model-

[8] Only one-tenth of the personal and friends corpus was used in these preliminary experiments.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2016.2635445, IEEE/ACM Transactions on Audio, Speech, and Language Processing

and feature-based personalization respectively. For sentence-dependent (SD) features, we viewed the 1000-best list as a single document and used the LDA topic model to infer the topic distribution, and then searched for the closest sentences as mentioned in Section VI-B to construct the sentence-dependent feature of each utterance for rescoring. For user-dependent (UD) features in the proposed approach, because the feature is extracted from the personal corpus of the user and is independent of the input utterance, the feature extraction process does not depend on ASR.

In Table IV, when the personal data ($\mathcal{B}$+$\mathcal{S}$) was involved, the LMs personalized by the model-based approach outperformed the universal RNNLM (rows (c-1) v.s. (b)). For the model-based approach, further involving the friend corpora was slightly helpful (rows (c-2) v.s. (c-1)). To our surprise, although the sentence-dependent feature-based approach reduced the PPL a lot in Table I, it was not very helpful for WER. It was even worse than the universal RNNLM in some cases (rows (d-3) v.s. (b) for 'h100' and 'h200'). This may be because for sentence-dependent feature, the topic distribution from the n-best list was inaccurate due to ASR errors. For 200 hidden layer units the proposed approach with user-dependent feature is better than sentence-dependent feature in terms of WER (rows (d-1) v.s. (d-3, 4, 5) for 'h200'). This may be because the user-dependent feature is estimated from the training corpus of target user and thus is not influenced by ASR errors at all.

In order to verify the hypothesis in the last paragraph, we did the oracle experiments in section (e), in which we used the topic distribution of the reference transcription of the utterance to replace the topic distribution of n-best list and for use in rescoring[9]. As expected, with topic distributions from the reference transcriptions, the results of sentence-dependent features improved in all cases (rows (e-1) v.s. (d-3), (e-2) v.s. (d-4), (e-3) v.s. (d-5)). With the reference transcriptions, the sentence-dependent feature (SD) is better in most cases than user-dependent feature (UD) (section (e) v.s. rows (d-1) and (d-2), except for row (e-1) in 'h100'). This verifies that ASR errors did influence the extraction of the sentence-dependent features. Since the extraction of user-dependent features is not affected by ASR errors, it can be useful when the ASR results are relatively poor.

In section (f), we further integrated the model-based approach in section (c) and the feature-based approach in section (d). To integrate the two approaches in n-best list rescoring, both approaches were used to compute the scores of each path, and then the scores from the two approaches were averaged to obtain the final score[10]. Row (f-1) is the results integrating model-based and feature-based approaches using personal corpora (rows (c-1) + (d-3)), while in row (f-2), both personal and friends corpora were involved (rows (c-2) + (d-4)). In row (f-3), the feature-based approach was further improved by the topic distribution of the sentence in question (rows (c-2) + (d-5)). All the results obtained by integration show improvements over individuals (rows (f-1) v.s. (c-1), (d-3), rows (f-2) v.s. (c-2), (d-4) and rows (f-3) v.s. (c-2), (d-5)). The results suggest that the model-based and feature-based approaches are complementary to each other. The best result obtained by integration (39.95% in (f-3)) yielded 0.24% WER reduction compared to the best result by the model-based personalization (40.19% in (c-2)) and 0.20% WER reduction compared to the best result by the feature-based personalization (40.15% in (d-5)).

TABLE IV: Word error rate (WER) results with same notation as in Table I. The GMM-based AMs was used here. For sentence-dependent (SD) features, topic distributions are estimated from the n-best lists in section (d), and from the reference transcriptions in section (e) (Oracle). The model-based approach in section (c) and the feature-based approach in section (d) were integrated in section (f) (Intgr). The superscript labels $^\alpha$ indicate significantly better than the non-personalized RNNLM in row (b) in terms of the pair-wise t-test with significance level at 0.05.

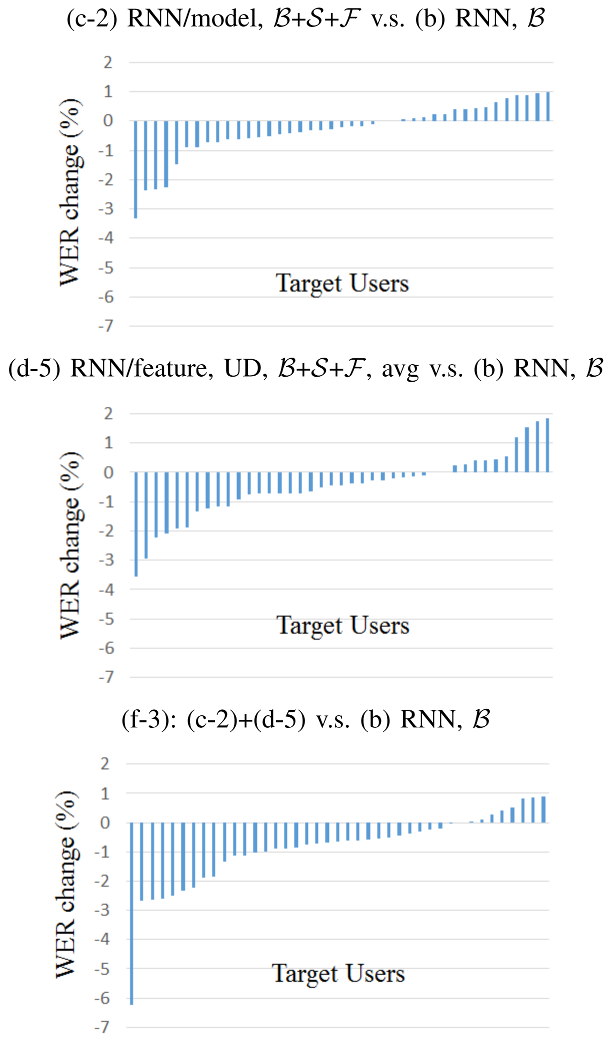
Word Error Rate (%)		h50	h100	h200
(a)	(a-1) KN3, $\mathcal{B}$	43.80		
	(a-2) KN3, $\mathcal{B}$+$\mathcal{S}$	43.39		
	(a-3) KN3, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$	41.95		
(b)	RNN, $\mathcal{B}$	40.52	40.51	40.31
(c)	(c-1) RNN/model, $\mathcal{B}$+$\mathcal{S}$	40.42	40.31^α	40.20
	(c-2) RNN/model, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$	40.40	40.30^α	40.19
(d)	(d-1) RNN/feature, UD, $\mathcal{B}$+$\mathcal{S}$	40.48	40.31	40.16^α
	(d-2) RNN/feature, UD, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$	40.64	40.29^α	40.32
	(d-3) RNN/feature, SD, $\mathcal{B}$+$\mathcal{S}$	40.47	40.54	40.36
	(d-4) RNN/feature, SD, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$	40.43	40.36	40.40
	(d-5) RNN/feature, SD, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$, avg	40.23^α	40.15^α	40.26
(e) Oracle	(e-1) RNN/feature, SD, $\mathcal{B}$+$\mathcal{S}$	40.15^α	40.32	40.09
	(e-2) RNN/feature, SD, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$	40.03^α	40.05^α	39.95
	(e-3) RNN/feature, SD, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$, avg	39.40^α	39.33^α	39.45^α
(f) Intgr	(f-1): (c-1) + (d-3)	40.28^α	40.19^α	39.99^α
	(f-2): (c-2) + (d-4)	40.27^α	40.23^α	40.11^α
	(f-3): (c-2) + (d-5)	40.15^α	39.95^α	39.98^α

In conclusion, for the real best result obtained by personalization in Table IV (39.95% in (f-3)), we reduced WER by 2.00% compared to the best KN3 model with the friends corpora (41.95% in (a-3)), from which the 1000-best lists for rescoring were obtained. Compared to the non-personalized RNNLM, personalization reduced WER by 0.36% (39.95% in (f-3) v.s. 40.31% in (b)). In the oracle case the results can be better than integrating the two personalization approaches (39.33% in (e-3) v.s. 39.95% in (f-3)), indicating the room for further improvement.

2) WER over all target users: Because the average does not reveal whether the proposed approach is actually helpful for most users or just for a small subset of users, in Fig. 4 we plot in addition the WER changes obtained across the all 42 target users. The three figures from the top to the bottom are respectively the WER changes by the three different personalization approaches: model-based personalization approach (RNN/model, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$ in row (c-2) in Table IV), sentence-dependent features (RNN/feature, SD, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$, avg, in row

[9] In the oracle experiments in section (e), user-dependent results were the same as those in (d-1) and (d-2) because ASR was not involved in extracting user-dependent features.

[10] Equal weights were given to the two personalization approaches here.


(c-2) RNN/model, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$ v.s. (b) RNN, $\mathcal{B}$

(d-5) RNN/feature, UD, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$, avg v.s. (b) RNN, $\mathcal{B}$

(f-3): (c-2)+(d-5) v.s. (b) RNN, $\mathcal{B}$

Fig. 4: WER changes across all 42 target users. The three figures from the top to the bottom are respectively the WER changes by the three different personalization approaches: model-based personalization approach (RNN/model, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$ in row (c-2) in Table IV), sentence-dependent features (RNN/feature, SD, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$, avg, in row (d-5)) and their integration ((c-2) + (d-5) in row (f-3)). The number of hidden layer units in the figures was 50.

(d-5)) and their integration ((c-2) + (d-5) in row (f-3)). The number of hidden layer units in the figures was 50. Each figure has 42 bars for the 42 target users, sorted based on the WER change. Here a negative value means that the feature-based personalization yielded a WER reduction for the user as compared with other approaches.

From the top and middle figures in Fig. 4, we found that both model-based and feature-based personalization reduced the WER for much more than half of the target users. The model-based personalization had worse WERs for 15 users, while the feature-based personalization only had 10, and all other users had WER reductions. On the other hand, for model-based personalization, all the users had worse WERs by less than 1.0%, whereas 4 target users had more than 2.0% WER increase for the feature-based approach. Compared the WER changes of the model- and feature-based personalization, we found that each approach has its strong point. For the model-based personalization, even when it increased WER, the degree of increase was limited; while the feature-based personalization hurt the WER for less users than the model-based approach. This explains the reason why integrating the two approaches can be very helpful. The WER change for integrating the two approaches is shown in the bottom figure of Fig. 4, which had obviously better performance than the top and middle figures. When integrating the two approaches, only 8 target users had worse WERs, and all by less than 1.0%.

TABLE V: Word error rate (WER) results with the same notation as in Table IV. The DNN-based AMs were used here.

Word Error Rate (%)		h50
(a)	KN3, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$	35.27
(b)	RNN, $\mathcal{B}$	33.75
(c)	(c-1) RNN/model, $\mathcal{B}$+$\mathcal{S}$	33.60
	(c-2) RNN/model, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$	33.63
(d)	(d-1) RNN/feature, UD, $\mathcal{B}$+$\mathcal{S}$	33.64
	(d-2) RNN/feature, UD, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$	33.64
	(d-3) RNN/feature, SD, $\mathcal{B}$+$\mathcal{S}$	33.69
	(d-4) RNN/feature, SD, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$	33.69
	(d-5) RNN/feature, SD, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$, avg	33.37
(e) Oracle	(e-1) RNN/feature, SD, $\mathcal{B}$+$\mathcal{S}$	33.45
	(e-2) RNN/feature, SD, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$	33.29
	(e-3) RNN/feature, SD, $\mathcal{B}$+$\mathcal{S}$+$\mathcal{F}$, avg	32.84

3) Significance Tests: Statistical significance tests for the WER results were performed. Here the WER for each user was considered as a sample, and the pair-wised t-test with the significant level at 0.05 was used to test the significance. The results are shown in Table IV. The superscript labels $^{\alpha}$ indicate significantly better than the non-personalized RNNLM in row (b). Both model- (section (c)) and feature-based personalization (section (d)) did not always improve the non-personalized RNNLM significantly. However, integrating the two approaches (section (f)) yielded significant improvements over the non-personalized RNNLM in all the cases.

B. DNN-based Acoustic Models

Table V reports the WERs obtained by the DNN-based AMs with the same notation as in Table IV. Section (a) is the WER for the personalized trigram LMs. Compared the results in Tables IV and V, the DNN-based AMs yielded much lower WERs than the GMM-based ones (35.27% in the row (a) of Table V v.s. 41.95% in the row (a-3) of Table IV). Section (b) shows rescoring results using RNNLM without personalization, and section (c) is for model-based personalization. Sections (d) and (e) were for feature-based personalization. For sentence-dependent (SD) features, topic distributions were estimated from the n-best lists in section (d), and from the reference transcriptions in section (e). Even though DNN-based AMs remarkably reduced WERs, from Table V we found that personalization is still helpful (sections (c),(d),(e) v.s. (b)).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2016.2635445, IEEE/ACM Transactions on Audio, Speech, and Language Processing

X. Conclusions

In this paper, we investigate RNNLM personalization using data crawled over social networks. We explore two RNNLM personalization methods: model-based and feature-based. In the model-based approach, the RNNLM parameters are fine-tuned for each user, yielding an RNNLM model for each user. The feature-based approach is based on a user characteristic feature extracted from the user corpus and friends corpora, which can be not only user-dependent but sentence-dependent. With the user characteristic feature, a universal RNNLM predicts different word distributions for different users given the same context. Experiments demonstrated good improvements in both perplexity and WER for both personalization methods. Moreover, we find that integrating the two personalization methods is helpful for WER reduction.

References

[1] M. Speretta and S. Gauch, “Personalized search based on user search histories,” in *Proc. on Web Intelligence*, 2005.

[2] X. Shen, B. Tan, and C. Zhai, “Implicit user modeling for personalized search,” in *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, 2005.

[3] G.-R. Xue, J. Han, Y. Yu, and Q. Yang, “User language model for collaborative personalized search,” *ACM Trans. Inf. Syst.*, vol. 27, no. 2, pp. 11:1–11:28, Mar. 2009.

[4] P. A. Chirita, C. S. Firan, and W. Nejdl, “Personalized query expansion for the web,” in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007.

[5] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *Computer*, 2009.

[6] F. Walter, S. Battiston, and F. Schweitzer, “A model of a trust-based recommendation system on a social network,” *Autonomous Agents and Multi-Agent Systems*, 2008.

[7] M.-H. Park, J.-H. Hong, and S.-B. Cho, “Location-based recommendation system using bayesian users preference model in mobile devices,” in *Ubiquitous Intelligence and Computing*, 2007.

[8] P.-H. Su, C.-H. Wu, and L.-S. Lee, “A recursive dialogue game for personalized computer-aided pronunciation training,” *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 23, no. 1, pp. 127–141, Jan 2015.

[9] Y. H. Cho, J. K. Kim, and S. H. Kim, “A personalized recommender system based on web usage mining and decision tree induction,” *Expert Systems with Applications*, 2002.

[10] Y.-Y. Huang, R. Yan, T.-T. Kuo, and S.-D. Lin, “Enriching cold start personalized language model using social network information,” *ACL*, pp. 611–617, 2014.

[11] G.-R. Xue, J. Han, Y. Yu, and Q. Yang, “User language model for collaborative personalized search,” *ACM Transactions on Information Systems (TOIS)*, vol. 27, no. 2, p. 11, 2009.

[12] A. Younus, C. ORiordan, and G. Pasi, “A language modeling approach to personalized search based on users microblog behavior,” in *Advances in Information Retrieval*. Springer, 2014, pp. 727–732.

[13] R. Kuhn and R. De Mori, “A cache-based natural language model for speech recognition,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 6, pp. 570–583, 1990.

[14] R. M. Iyer and M. Ostendorf, “Modeling long distance dependence in language: Topic mixtures versus dynamic cache models,” *Speech and Audio Processing, IEEE Transactions on*, vol. 7, no. 1, pp. 30–39, 1999.

[15] R. Rosenfeld, “A maximum entropy approach to adaptive statistical language modelling,” *Computer Speech & Language*, vol. 10, no. 3, pp. 187–228, 1996.

[16] J. R. Bellegarda, “Statistical language model adaptation: review and perspectives,” *Speech Communication*, 2004.

[17] A. Heidel and L.-S. Lee, “Robust topic inference for latent semantic language model adaptation,” in *Proc. on ASRU*, 2007.

[18] H. Bo-June and J. Glass, “Style and topic language model adaptation using HMM-LDA,” in *Proc. on EMNLP*, 2006.

[19] T. Yik-Cheung and T. Schultz, “Unsupervised language model adaptation using latent semantic marginals,” in *Proc. on InterSpeech*, 2006.

[20] Y.-C. Tam and T. Schultz, “Correlated latent semantic model for unsupervised LM adaptation,” in *ICASSP*, 2007.

[21] T.-H. Wen, H.-Y. Lee, and L.-S. Lee, “Personalized language modeling by crowd sourcing with social network data for voice access of cloud applications,” in *Proc. on SLT*, 2012.

[22] T.-H. Wen, A. Heidel, H.-Y. Lee, Y. Tsao, and L.-S. Lee, “Recurrent neural network based language model personalization by social network crowdsourcing.” in *InterSpeech*, 2013, pp. 2703–2707.

[23] B.-H. Tseng, H.-Y. Lee, and L.-S. Lee, “Personalizing universal recurrent neural network language model with user characteristic features by social network crowdsourcing,” in *ASRU*, 2015.

[24] D. Hakkani-Tur, G. Tur, and L. Heck, “Research challenges and opportunities in mobile applications,” *Signal Processing Magazine, IEEE*, 2011.

[25] C. J. Leggetter and P. C. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models,” *Computer Speech and Language*, 1995.

[26] P. C. Woodland, “Speaker adaptation for continuous density hmms: A review,” in *Proc. on ITRW on Adaptation Methods for Speech Recognition*, 2001.

[27] l. G. Jean and C.-H. Lee, “Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains,” *IEEE Transactions on Speech and Audio Processing*, 1994.

[28] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[29] A. Doan, R. Ramakrishnan, and A. Y. Halevy, “Crowdsourcing systems on the world-wide web,” *Communications of the ACM*, 2011.

[30] Munro and Robert, “Crowdsourcing and language studies: the new generation of linguistic data,” in *Proc. on NAACL*, 2010.

[31] K. Berberich, S. Bedathur, O. Alonso, and G. Weikum, “A language modeling approach for temporal information needs,” in *Advances in Information Retrieval*, 2010.

[32] J. Liu, S. Cyphers, P. Pasupat, I. McGraw, and J. Glass, “A conversational movie search system based on conditional random field,” in *Proc. on InterSpeech*, 2012.

[33] I. McGraw, S. Cyphers, P. Pasupat, J. Liu, and J. Glass, “Automating crowd-supervised learning for spoken language systems,” in *Proc. on InterSpeech*, 2012.

[34] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, “Class-based n-gram models of natural language,” *Computational Linguistics*, 1992.

[35] A. Stolcke, “Entropy-based pruning of backoff language models,” in *Proc. of the DARPA Broadcast News Transcription and Understanding Workshop*, 2000.

[36] R. Iyer and M. Ostendorf, “Modeling long distance dependence in language: topic mixtures versus dynamic cache models,” *IEEE Transactions on Speech and Audio Processing*, 1999.

[37] A. Heidel, H.-A. Chang, and L.-S. Lee, “Language model adaptation using latent dirichlet allocation and an efficient topic inference algorithm,” in *Proc. on InterSpeech*, 2007.

[38] M. Federico, “Efficient language model adaptation through mdi estimation,” in *Proc. on EuroSpeech*, 1999.

[39] C. Chelba and F. Jelinek, “Structured language modeling,” *Computer Speech and Language*, 2000.

[40] J. Wu and S. Khudanpur, “Combining nonlocal, syntactic and n-gram dependencies in language modeling,” in *Proc. on EuroSpeech*, 1999.

[41] S. Khudanpur and J. Wu, “Maximum entropy techniques for exploiting syntactic, semantic and collocational dependencies in language modeling,” *Computer Speech and Language*, 2000.

[42] H. S. Le, A. Allauzen, and Y. Fran, “Measuring the influence of long range dependencies with neural network language models,” in *Proc. on NAACL-HLT Workshop*, 2012.

[43] I. Oparin, M. Sundermeyer, H. Ney, and J. Gauvain, “Performance analysis of neural networks in combination with n-gram language models,” in *Proc. on ICASSP*, 2012.

[44] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, “A neural probabilistic language model,” *Journal of Machine Learning Research*, 2003.

[45] J. Park, X. Liu, M. J. F. Gales, and P. C. Woodland, “Improved neural network based language modeling and adaptation,” in *Proc. on InterSpeech*, 2010.

[46] H.-S. Le, I. Oparin, A. Allauzen, J.-L. Gauvain, and F. Yvon, “Structured output layer neural network language model,” in *Proc. on ICASSP*, 2011.

[47] X. Liu, M. J. F. Gales, and P. C. Woodland, "Improving lvcsr system combination using neural network language model cross adaptation," in *Proc. on InterSpeech*, 2011.

[48] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. on InterSpeech*, 2010.

[49] T. Mikolov, S. Kombrink, L. Burget, J. H. Černockỳ, and S. Khudanpur, "Extensions of recurrent neural network language model," in *ICASSP*. IEEE, 2011, pp. 5528–5531.

[50] T. Mikolov and G. Zweig, "Context dependent recurrent neural network language model." in *SLT*, 2012, pp. 234–239.

[51] M. Sundermeyer, H. Ney, and R. Schlüter, "From feedforward to recurrent LSTM neural networks for language modeling," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 23, no. 3, pp. 517–529, 2015.

[52] X. Liu, X. Chen, Y. Wang, M. J. F. Gales, and P. C. Woodland, "Two efficient lattice rescoring methods using recurrent neural network language models," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 24, no. 8, pp. 1438–1449, 2016.

[53] X. Chen, X. Liu, Y. Wang, M. J. F. Gales, and P. C. Woodland, "Efficient training and evaluation of recurrent neural network language models for automatic speech recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 24, no. 11, pp. 2146–2157, 2016.

[54] Y. Shi, P. Wiggers, and C. M. Jonker, "Towards recurrent neural networks language models with linguistic and contextual features," in *Proc. on InterSpeech*, 2012.

[55] M. L. Seltzer and J. Droppo, "Multi-task learning in deep neural networks for improved phoneme recognition," in *ICASSP*, 2013.

[56] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *ASRU*. IEEE, 2013, pp. 55–59.

[57] V. Gupta, P. Kenny, P. Ouellet, and T. Stafylakis, "I-vector-based speaker adaptation of deep neural networks for french broadcast audio transcription," in *ICASSP*. IEEE, 2014, pp. 6334–6338.

[58] A. K. McCallum, "MALLET: A machine learning for language toolkit," 2002, http://mallet.cs.umass.edu.

[59] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[60] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proc. on Spoken Language Processing*, 2002.

[61] F. James, "Modified kneser-ney smoothing of n-gram models," *Research Institute for Advanced Computer Science, Tech. Rep. 00.07*, 2000.

[62] T. Mikolov, S. Kombrink, A. Deoras, L. Burget, and J. Cernocky, "RNNLM - recurrent neural network language modeling toolkit," in *Proc. on ASRU*, 2011.

[63] C.-F. Yeh, A. Heidel, H.-Y. Lee, and L.-S. Lee, "Recognition of highly imbalanced code-mixed bilingual speech with frame-level language detection based on blurred posteriorgram," in *ICASSP*. IEEE, 2012, pp. 4873–4876.

[64] H.-Y. L. amd Yueh-Lien Tang, H. Tang, and L.-S. Lee, "Spoken term detection from bilingual spontaneous speech using code-switched lattice-based structures for words and subword units," in *ASRU*, 2009.

[65] C.-Y. Tseng and T. Visceglia, "AESOP (Asian English Speech Corpus Project) and TWNAESOP," in *International Conference and Workshop on TEFL & Applied Linguistics*, 2010.

Hung-yi Lee received the M.S. and Ph.D. degrees from National Taiwan University (NTU), Taipei, Taiwan, in 2010 and 2012, respectively. From September 2012 to August 2013, he was a postdoctoral fellow in Research Center for Information Technology Innovation, Academia Sinica. From September 2013 to July 2014, he was a visiting scientist at the Spoken Language Systems Group of MIT Computer Science and Artificial Intelligence Laboratory (CSAIL). He is currently an assistant professor of the Department of Electrical Engineering of National Taiwan University, with a joint appointment at the Department of Computer Science & Information Engineering of the university. His research focuses on spoken language understanding, speech recognition and machine learning.

Bo-Hsiang Tseng was born in Taipei, Taiwan. He earned a Master degree, under the supervision of Professor Lin-shan Lee in Speech Processing Lab, in the Graduate Institute of Communication Engineering from National Taiwan University (NTU) in 2016. He also holds B.S. degree in Electrical and Computer Engineering from National Chiao Tung University (NCTU). His research interests include deep learning, speech recognition (esp. language modeling), spoken language understanding (SLU), natural language processing (NLP) and machine learning.

Tsung-Hsien Wen is a PhD student in Dialogue Systems Group, University of Cambridge, United Kingdom. He received both his B.S. and M.S. degrees in Electrical Engineering from National Taiwan University, Taipei, Taiwan. His research focuses on language generation and end-to-end dialogue modelling, specifically in learning to generate responses for task-oriented dialogue systems. He was the tutor of the "Deep Learning and NLG" tutorial at INLG 2016 and has given invited seminars to research groups including Google HQ, Xerox Research Centre Europe, and Baidu China. He has published more than 20 peer-reviewed conference papers and received best paper awards at both EMNLP 2015 and SigDial 2015. He is a current member of Darwin College.

Yu Tsao received the B.S. and M.S. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1999 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from Georgia Institute of Technology, Atlanta, GA, USA, in 2008. From 2009 to 2011, he was a Researcher with the National Institute of Information and Communications Technology, Kyoto, Japan, where he was engaged in research and product development in automatic speech recognition for multilingual speech-to-speech translation. He is currently an Associate Research Fellow with the Research Center for Information Technology Innovation, Academia Sinica, Taiwan. His research interests include speech recognition, audio-coding, deep neural networks, biosignals, and acoustic modeling.