Knowledge Transfer Using User-Generated Data within Real-Time Cloud Services

Jing Zhang[1*], Jianhan Pan[2], Zhicheng Cai[1], Min Li[1], Lin Cui[3],
[1] School of Computer Science and Engineering, Nanjing University of Science and Technology
Nanjing, Jiangsu 210094 - China
[e-mail: jzhang, caizhicheng, ml025@njust.edu.cn]
[2] School of Computer Science and Technology, Jiangsu Normal University
Xuzhou, Jiangsu 221116 - China
[e-mail: jhpan@xznu.edu.cn]
[3] Intelligent Information Processing Laboratory, Suzhou University
Suzhou, Anhui 221116 - China
[e-mail: lcui@ahszu.edu.cn]
*Corresponding author: Jing Zhang

Received April 25, 2019; revised August 7, 2019; accepted September 16, 2019; published January 31, 2020

Abstract

When automatic speech recognition (ASR) is provided as a cloud service, it is easy to collect voice and application domain data from users. Harnessing these data will facilitate the provision of more personalized services. In this paper, we demonstrate our transfer learning-based knowledge service that built with the user-generated data collected through our novel system that deliveries personalized ASR service. First, we discuss the motivation, challenges, and prospects of building up such a knowledge-based service-oriented system. Second, we present a Quadruple Transfer Learning (QTL) method that can learn a classification model from a source domain and transfer it to a target domain. Third, we provide an overview architecture of our novel system that collects voice data from mobile users, labels the data via crowdsourcing, utilises these collected user-generated data to train different machine learning models, and delivers the personalised real-time cloud services. Finally, we use the E-Book data collected from our system to train classification models and apply them in the smart TV domain, and the experimental results show that our QTL method is effective in two classification tasks, which confirms that the knowledge transfer provides a value-added service for the upper-layer mobile applications in different domains.

Keywords: Cloud computing, distributed computing, personalized service, transfer learning, user behavior mining

1. Introduction

To improve the efficiency of information input in mobile devices, many intelligent mobile devices currently trend to use multimodal human-computer interaction technology, among which speech interaction plays an important role. It not only provides a more natural way of human-machine control but also allows the system to accumulate a variety of user-generated data. Based on these data, we can extract and learn some knowledge and provide the knowledge as a service. However, the achievement of this ambitious goal is full of challenges. As we know, recognising continuous free-speech audio streams requires massive computing resources such as a large memory to store trained language models and a high-performance CPU to run complicated decoding algorithms, which usually exceeds the capability of a single mobile device. Thanks to the growing bandwidth of the wireless Internet, mobile applications can contact public cloud services to extend their computational abilities. For example, the Apple iOS system integrates an automatic voice assistant application, namely Siri[1], to help users arrange their daily lives. Users with Siri and active mobile Internet connections speak out what they want to do, then their voices flow to a remote automatic speech recognition (ASR) service, and then the corresponding recognized results are immediately sent back to drive the application to take specific actions such as sending messages, scheduling meetings, placing phone calls, retrieving information from Internet, and so on. Meanwhile, the ASR-translated texts in the system form a huge data source, from which various knowledge could be mined through machine learning methods.

With the recent development of cloud computing and machine learning technology, in China, many artificial intelligence companies, such as Baidu, Tencent, and iFlyTek, have built up large-scale automatic speech recognition service for open access in the past several years, delivering their speech and language processing technologies to the public. These services offer mobile applications the ability to accept Chinese Mandarin speeches as input data and obtaining their corresponding textual contents from remote hosts. Collaborating with iFlytek, after five-year construction, we have built up three experimental data centres all over China with about 1,000 high-performance servers to deliver ASR cloud service. The total volume of the system is about 14 petabytes, and the incremental daily user data is up to 20 terabytes. Our speech cloud service covers about 400 million people with tremendous different featured mobile applications such as Input Method Editor, automobile navigator, personal assistant, smart TV, standard Chinese language education, and so on.

Since we have already accumulated huge amonts of user-generated data through our ASR cloud service, we expect that these data can be used for the evaluation of our ASR cloud system and provide a value-adding knowledge service for various applications that are built upon the top of our system. Thus, we have been designing a novel system for both real-time personalised service delivery and incoming data analysis for knowledge supply. In recent year, many ASR systems have been set up for different purposes with one basic function of translating speeches to texts [1, 17, 18]. However, only a few of them addressed the personalised issue through training different language models [11]. With accumulated user

[1] https://www.apple.com/ios/siri/

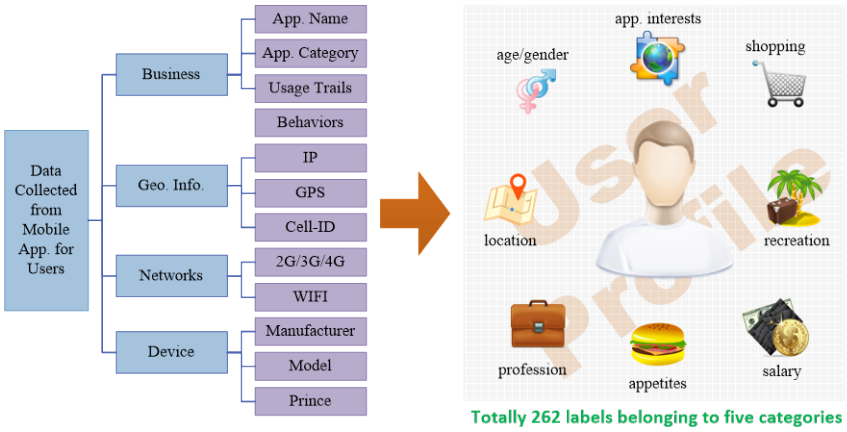


Fig. 1. User data are collected from the applications that integrate the client APIs of our ASR cloud service. We use these data to create user profile for each user in the system.

speech data, no previous work tried to utilise them for some other applications. Our work makes the first step to these issues. The contributions of this paper are three-fold. (1) We demostrate that knowledge service can be achieved from the ASR-translated user-generated data using transfer learning technology. (2) A set of state-of-the-art distributed systems and tools are used to build an entire eco-system, where user data can be collected and used for the self-evolution of the systems. (3) We demonstrate how the crowdsourcing plays in this human-machine hybrid system to facilitate the speed of model reproduction.

The remainder of the paper is organized as follows. In section 2, we analyse the challenges that we may face in building the personalized ASR and knowledge services in real-time cloud systems. Section 3 provides a knowledge transfer learning method that can share classification models between two application domains. Section 4 provides an overview of the architecture of our system that provides both ASR and knowledge services. Section 5 shows that the experimental results of our method. Section 6 concludes the paper with future work.

2. Motivations and Challenges

In this section, we discuss prospective features of our ongoing real-time service-oriented system and the challenges that we may face while achieving the desired features.

2.1 Personalized Services

It would be much easier to collect users' voice data through ASR as a cloud service, which provides us great opportunities to provide more personalised services. We are fond of personalisation for two reasons. From the aspect of commercial interests, under recently rapid growing mobile commercial environments, service subscribers' intention has already switched to those providers whose services are tailored for individuals [7]. From the other aspect of technology, personalised services can achieve higher recognition accuracy. It is known that the Chinese language is one of the most complicated languages in the world. China is basically

divided into eight dialect areas. People from different dialect areas not only have different accents, but also may have different expressions when describing the same object, a different order of words when forming a sentence, and different meanings when using the same word. In an extreme case, people from in different districts of the same county may have very different accents. Additionally, people with different educational and professional backgrounds may have different language habits. Generally, the current ASR system requires users to speak a standard Mandarin language; although they try to do so when using mobile applications, certain users can hardly eliminate the impact of their dialects and language habits. It would be much better to build an acoustic and language model for each. The challenge here is that although we can easily collect users' voice data, we must find an efficient way of labelling them with relatively low cost, and then utilise these labelled data to train acoustic and language models.

We are going to provide two kinds of personalised services. The first is our basic large-scale automatic speech recognition service. Different from the previous generic one which offers common models based on the standard Chinese Mandarin for all users, the novel system will provide specific acoustic and language models for each to make the system adaptations for the user's accent and idiomatic expression. As the most primary business of the company, the primary functions of this service have been implemented. The subsequent work includes improving the performance of these models, which is a research issue related to pattern recognition domain instead of data engineering and data mining. The second personalised service is sharing the knowledge discovered from these user-generated data including voices, transferred texts, and domain-related business data. Although we are still in the initial stage of providing knowledge service, we have made the first step. We utilise these data to build machine learning models for a cross-domain usage.

2.2 Real-time Service Delivery Services

Real-time service delivery is another essential requirement for mobile applications. To shorten the response time of interactions under current 3G/4G mobile communication environments, iFlyTek has set up three data centres in Beijing (northern China) with a 300-node cluster, Hefei (middle China) with a 400-node cluster and Guangzhou (southern China) with a 300-node cluster. The client API library of our speech cloud service integrated into a mobile device periodically measures the Round-Trip Time (RTT) to these data centres and selects the optimal one with the minimum RTT for service requests. Under a typical network condition, the maximum RTT of recognising a sentence with dozens of words does not exceed two seconds. After adopting a personalised service scheme, retrieving and loading specific user's model data from a repository with hundreds of millions of user-specific models within several hundred milliseconds becomes a significant challenge. Within a data centre, we must retrieve and load the models within the shortest possible time. Once a user travels from one place to another, resulting in switching the data centre, we must schedule this user's model data across different data centres if they cannot be found in the data centre that the user is currently located.

2.3 Knowledge Sharing

We deem our personalised ASR service as a kind of Platform as a Service (PaaS), which means it provides a basic input method service for all upper-layer mobile applications. A

person may use different mobile applications that all integrate the client library of our service. We can imagine the following scenes: In the morning, a businessman drives his car and speaks out where he wants to go towards a GPS navigation system. During his meetings, he records what he said using his digital recording pen, and then transfers these voices into texts so that they can be easily copied and pasted into his report and distribute them among his colleagues. At night, he watches his smart TV, which lets him speak out his favourite programs and make comments to the programs. All these applications require the ASR service. With these different applications, we can not only collect the input voice data but also obtain a lot of domain-specific information. Currently, it is easy for us to create a user profile for each user in the system from the data generated by different upper-layer applications as **Fig. 1** shows. A user profile in our system, each of which contains 262 labels belonging to five categories (such as demographic property, stage of life, an application preference, interests, and behaviour tendency), is a kind of knowledge commonly shared. However, compared with the simplicity of building up user profiles, it would be great challenges to fuse these heterogeneous data, to mine more profound potential knowledge behind it [16], and to share the knowledge among different mobile applications. The proposed proto-system starts from a mash-up of mixed data into service and gradually evolves to a stage of providing knowledge as a service.

3. Knowledge Transferring between Application Domains

In this section, we present our method that shares the knowledge discovered from user data across different application domain via transfer learning technology.

3.1 Background and Motivation

Through personalised ASR service, we plan to provide another kind of knowledge-based data services based on mining the data generated by users. In a real-world situation, user profiles are relatively simple while more useful information must be extracted from the incoming data. Compared with superficial information that could be obtained through simple statistical methods, we are interested in mining more profound knowledge using sophisticated techniques. In this section, we demonstrate our preliminary work on transferring knowledge from one application domain to another.

Since a real-time cloud service often provides a set of open APIs for developers to integrate our services in their applications, it would be very interesting that knowledge discovered from a mature application can benefit newly developed applications. After a three-year operation of our ASR cloud service in China, we have accumulated enough users' voice data and their corresponding transferred texts from the smart TV application. The collected data include the information about TV plays or movies that the users have viewed and the comments provided by the users. Then, we (or those annotators from crowdsourcing systems) labelled them and built a model to classify user interests. Of course, we can use this model to classify a user who begins to use a voice-controlled smart TV. Furthermore, this model can also provide additional value to some newly developed applications, for example, an E-Book application. The E-Book application is a virtual library for mobile users. Users can buy or rent e-books online, read them, make notes, and provide comments. Usually, a recommendation system is an essential part of such an intelligent application, which provides the book information that readers may be interested in based on their historical information of readings. The simplest

Table 1. Four Kinds of Concepts across Domains [12]

Notations		Descriptions
Shared Concept	Identical Concept	A concept is identical when it has the same extension and the same intension across domains.
	Synonymous Concept	A concept is synonymous when it has the different extension but the same intension across domains.
Non-shared Concept	Different Concept	A concept is different when it has the different extension and the different intension across domains.
	Ambiguous Concept	A concept is ambiguous when it has the same extension but the different intension across domains.

recommendation approach is to classify the users into different groups with different interests. For example, at least we shall have two attitudes towards an object: negative and positive. In the early days after the E-Book application was released into the market, we could only obtain some unlabeled data such as the books (or their abstracts) that users have browsed, and some comments provided. On the one hand, we do not have so many data because the accumulation of users takes a certain amount of time. On the other hand, without a clear profit prospect at the beginning stage of a product, the investor (a company or an individual developer) of E-Book may have no investment plan of labelling these data and building new models for its recommendation system. The recommendation system faces a well-known problem of cold starting. We still hope that the off-the-shelf model derived from the smart TV can be applied to E-Book, transferring the users' interests from TV to books, because they are sufficiently similar. For example, the people now watching *The Big Bang Theory* (a TV serial) may be interested in the book *A Brief History of Time* written by Stephen Hawking. We can achieve this goal by applying transfer learning [13] across different domains.

3.2 Transfer Learning with QTL

First, we transfer user voices into texts using our ASR system. Since the sentences related to the descriptions of TV programs are usually short, we combine multiple sentences from one user into a document. Then, we use our proposed Quadruple Transfer Learning (QTL) [12] algorithm to conduct knowledge transfer between to domains.

In a transfer learning, we usually have two mappings from raw features (i.e., documents in this scenario) to final classes. The mapping of the raw features to high-level concepts is represented as the *Concept Extension*, whose formalization is rather difficult. The mapping of high-level concepts to final classes (i.e., labels) is represented as the *Concepts Intension*. Traditional transfer learning techniques usually have weak discriminations for the high-level concepts, which results in a negative transfer phenomenon [13]. For example, in a review document in the domain of science fiction, if we obtain a high-level concept "imaginative", the class of the document is probably positive, because it means that the fiction is interesting. However, if we transfer this high-level concept to the domain of scientific article review, we may get a wrong class label (positive), because in that domain imaginative may be a negative judgment which means unrealistic. We discriminate four different high-level concepts between the source and the target domains, which are shown in **Table 1**. Although these concepts have been widely used in previous studies, they are never integrated together for

classification. Our QTL based on the non-negative matrix tri-factorization models these four kinds of concepts simultaneously.

Table 2. Notations and descriptions

Notation	Description
r	Domain index
s	The number of source domains
t	The number of target domains
X_r	The feature-example co-occurrence matrix of domain r
m	The number of features
c	The number of classes
n_r	The number of examples in domain r
U	The matrix of feature clusters
H	The matrix of the association between feature clusters and example classes
V	The matrix of example classes
k_1	The number of identical concepts
k_2	The number of synonymous concepts
k_3	The number of different concepts
k_4	The number of ambiguous concepts

To clearly present our solution, **Table 2** lists the notations used in our paper. We assume that in the representation of the feature-example co-occurrence matrix $X_r \in R_+^{m\times n_r}$ (including n_r examples, each of which has m features), each element is non-negative. We define matrix UHV^T that associates the features of all examples and their classes through matrices $U^{m\times k}$, $H^{k\times c}$ and $V^{n\times c}$. Each element $U_{[i,j]}$ represents the probability that the i-th original feature belongs to the j-th feature cluster. Each element $V_{[i,j]}$ represents the probability that the i-th example belongs to the jth class. Each element $H_{[i,j]}$ represents the probability that the i-th feature cluster belongs to the j-th class. The non-negativematrixtri-factorization model is to minimize the difference between the original features matrix and UHV^T, which can be formulated as follows:

$$\min_{U,H,V\geq 0} \left\| X - UHV^T \right\|^2, \quad \text{s.t.} \sum_{i=1}^{m} U_{[i,j]} = 1, \sum_{j=1}^{c} H_{[i,j]} = 1, \sum_{j=1}^{c} V_{[i,j]} = 1. \tag{1}$$

In QTL, because we have four kinds of concepts, we divide U and H into four parts corresponding to the identical (1), synonymous (2), different (3) and ambiguous (4) concepts, respectively. Hence, we let $U = [U^1_{m\times k_1}, U^2_{m\times k_2}, U^3_{m\times k_3}, U^4_{m\times k_4}]$, where $k_1 + k_2 + k_3 + k_4 = k$ and $U^*_{m\times k_*}$ represents the feature clusters. Accordingly, we let $H = [H^1_{k_1\times c}, H^2_{k_2\times c}, H^3_{k_3\times c}, H^4_{k_4\times c}]^T$, where $H^*_{k_*\times c}$ represents the association between example classes and the concept. In addition, we extend the Eq. (1) to multiple domains. Finally, the objective function of QTL can be defined as follows:

$$\mathcal{L}=\sum_{r=1}^{s+t}\left\|X_r-U_rH_rV_r^T\right\|^2=\sum_{r=1}^{s+t}\left\|X_r-\left[U^1,U_r^2,U_r^3,U^4\right]\begin{bmatrix}H^1\\H^2\\H_r^3\\H_r^4\end{bmatrix}V_r^T\right\|^2. \tag{2}$$

Here, $V_r^T\in R_+^{n_r\times c}(s+1\le r\le s+t)$ also represents the classifier used in the target domain. To quantify the relationships among the original features, the feature clusters, and the classes, we add a constraint condition to U_r, H_r and V_r simultaneously, and reduce the optimization problem as follows:

$$\begin{aligned}\min_{U_r,H_r,V_r\ge 0}\mathcal{L}\quad \text{s.t.}\ &\sum_{j=1}^{k_1}U_{[i,j]}^1=1,\sum_{j=1}^{k_2}U_{r[i,j]}^2=1,\sum_{j=1}^{k_3}U_{r[i,j]}^3=1,\sum_{j=1}^{k_4}U_{[i,j]}^4=1,\\ &\sum_{j=1}^{c}H_{[i,j]}^1=1,\sum_{j=1}^{c}H_{[i,j]}^2=1,\ \sum_{j=1}^{c}H_{r[i,j]}^3=1,\sum_{j=1}^{c}H_{r[i,j]}^4=1,\ \ \sum_{j=1}^{c}V_{r[i,j]}=1.\end{aligned} \tag{3}$$

To solve this non-convex optimization problem, we obtain the partial derivative of the objective function with respect to each variable. The objective function will reach its local optimal by iteratively updating the variables as follows [12]:

$$U_{[i,j]}^1\leftarrow U_{[i,j]}^1\sqrt{\frac{\left[\sum_{r=1}^{s+t}X_rV_r(H^1)^T\right]_{[i,j]}}{\left[\sum_{r=1}^{s+t}A_rV_r(H^1)^T+B_rV_r(H^1)^T+C_rV_r(H^1)^T+D_rV_r(H^1)^T\right]_{[i,j]}}}, \tag{4}$$

$$U_{r[i,j]}^2\leftarrow U_{r[i,j]}^2\sqrt{\frac{\left[X_rV_r(H^2)^T\right]_{[i,j]}}{\left[A_rV_r(H^2)^T+B_rV_r(H^2)^T+C_rV_r(H^2)^T+D_rV_r(H^2)^T\right]_{[i,j]}}}, \tag{5}$$

$$U_{r[i,j]}^3\leftarrow U_{r[i,j]}^3\sqrt{\frac{\left[X_rV_r(H_r^3)^T\right]_{[i,j]}}{\left[A_rV_r(H_r^3)^T+B_rV_r(H_r^3)^T+C_rV_r(H_r^3)^T+D_rV_r(H_r^3)^T\right]_{[i,j]}}}, \tag{6}$$

$$U_{[i,j]}^4\leftarrow U_{[i,j]}^4\sqrt{\frac{\left[\sum_{r=1}^{s+t}X_rV_r(H_r^4)^T\right]_{[i,j]}}{\left[\sum_{r=1}^{s+t}A_rV_r(H_r^4)^T+B_rV_r(H_r^4)^T+C_rV_r(H_r^4)^T+D_rV_r(H_r^4)^T\right]_{[i,j]}}}, \tag{7}$$

$$H_{[i,j]}^1\leftarrow H_{[i,j]}^1\sqrt{\frac{\left[\sum_{r=1}^{s+t}(U^1)^TX_rV_r\right]_{[i,j]}}{\left[\sum_{r=1}^{s+t}(U^1)^TA_rV_r+(U^1)^TB_rV_r+(U^1)^TC_rV_r+(U^1)^TD_rV_r\right]_{[i,j]}}}, \tag{8}$$

$$H_{[i,j]}^2\leftarrow H_{[i,j]}^2\sqrt{\frac{\left[\sum_{r=1}^{s+t}(U_r^2)^TX_rV_r\right]_{[i,j]}}{\left[\sum_{r=1}^{s+t}(U_r^2)^TA_rV_r+(U_r^2)^TB_rV_r+(U_r^2)^TC_rV_r+(U_r^2)^TD_rV_r\right]_{[i,j]}}}, \tag{9}$$

$$H^3_{r[i,j]} \leftarrow H^3_{r[i,j]} \sqrt{\frac{\left[(U^3_r)^T X_r V_r\right]_{[i,j]}}{\left[(U^3_r)^T A_r V_r + (U^3_r)^T B_r V_r + (U^3_r)^T C_r V_r + (U^3_r)^T D_r V_r\right]_{[i,j]}}}, \tag{10}$$

$$H^4_{r[i,j]} \leftarrow H^4_{r[i,j]} \sqrt{\frac{\left[(U^4)^T X_r V_r\right]_{[i,j]}}{\left[(U^4)^T A_r V_r + (U^4)^T B_r V_r + (U^4)^T C_r V_r + (U^4)^T D_r V_r\right]_{[i,j]}}}, \tag{11}$$

$$V_{r[i,j]} \leftarrow V_{r[i,j]} \sqrt{\frac{\left[X_r{}^T U_r V_r\right]_{[i,j]}}{\left[V_r H_r{}^T U_r^T U_r H_r\right]_{[i,j]}}}, \tag{12}$$

where $A_r = U^1 H^1 V_r^T$, $B_r = U_r^2 H^2 V_r^T$, $C_r = U_r^3 H_r^3 V_r^T$, and $D_r = U^4 H_r^4 V_r^T$. In each matrix above, when all values of its elements have been calculated using Eqs. (4) ~ (12), they will be further normalized as follows:

$$\begin{aligned}
&U^1_{[i,j]} \leftarrow \frac{U^1_{[i,j]}}{\sum_{j=1}^{k_1} U^1_{[i,j]}},\ U^2_{r[i,j]} \leftarrow \frac{U^2_{r[i,j]}}{\sum_{j=1}^{k_2} U^2_{r[i,j]}},\ U^3_{r[i,j]} \leftarrow \frac{U^r_{r[i,j]}}{\sum_{j=1}^{k_3} U^3_{r[i,j]}},\\
&U^4_{[i,j]} \leftarrow \frac{U^4_{[i,j]}}{\sum_{j=1}^{k_4} U^4_{[i,j]}}, H^1_{[i,j]} \leftarrow \frac{H^1_{[i,j]}}{\sum_{j=1}^{c} H^1_{[i,j]}},\ , H^2_{[i,j]} \leftarrow \frac{H^2_{[i,j]}}{\sum_{j=1}^{c} H^2_{[i,j]}},\\
&H^3_{r[i,j]} \leftarrow \frac{H^3_{[i,j]}}{\sum_{j=1}^{c} H^3_{r[i,j]}},\ H^4_{[i,j]} \leftarrow \frac{H^4_{[i,j]}}{\sum_{j=1}^{c} H^4_{r[i,j]}}, V_{r[i,j]} \leftarrow \frac{V_{r[i,j]}}{\sum_{j=1}^{c} V_{r[i,j]}}
\end{aligned} \tag{13}$$

Finally, we summarize our transfer learning method in Algorithm QTL as follows. The computational complexity of QTL is $O(\sum_{r=1}^{s+t} \text{maxInteration} \cdot mkn_r)$.

Algorithm 1. Quadruple Transfer Learning (QTL)

Input: $\{X_r\}_{r=1}^{s+t}, \{V_r\}_{r=1}^{s}$, parameters k_1, k_2, k_3, k_4, and *maxIteration*

Output: $U^1, U_r^2, U_r^3, U^4, H^1, H^2, H_r^3, H_r^4, (1 \le r \le s+t)$ and $V_r (1+s \le r \le s+t)$

1. normalise the data matrices by $X_{r[i,j]} \leftarrow X_{r[i,j]} / \sum_{i=1}^{m} X_{r[i,j]}, (1 \le r \le s+t)$
2. $U^{4(0)}, H^{1(0)}, H^{2(0)}, H_r^{3(0)}, H_r^{4(0)}$ are randomly initialised, $U^{1(0)}, U_r^{2(0)}, U_r^{3(0)}$ are initialised with PLSA[8], , and $V_r^{(0)}$ is initialized by Logistic Regression.
3. **for** $k \leftarrow 1$ **to** *maxIteration* **do**
4. update $U^{1(k)}$ by Eq. (4)
5. **for** $r \leftarrow 1$ **to** $s+t$ **do** update $U_r^{2(k)}$ by Eq. (5) and $U_r^{3(k)}$ by Eq. (6)
6. update $U^{4(k)}$ by Eq. (7)
7. update $H^{1(k)}$ by Eq. (8) and $H^{2(k)}$ by Eq. (9)
8. **for** $r \leftarrow 1$ **to** $s+t$ **do** update $H_r^{3(k)}$ by Eq. (10) and $H_r^{4(k)}$ by Eq. (11)
9. **for** $r \leftarrow s+1$ **to** $s+t$ **do** update $V_r^{(k)}$ by Eq. (12).
10. normalise $U^{1(k)}, U_r^{2(k)}, U_r^{3(k)}, U^{4(k)}, H^{1(k)}, H^{2(k)}, H_r^{3(k)}, H_r^{4(k)}$ and $V_r^{(k)}$ by Eq. (13).
11. **return** $U^1, U_r^2, U_r^3, U^4, H^1, H^2, H_r^3, H_r^4, (1 \le r \le s+t)$ and $V_r (1+s \le r \le s+t)$

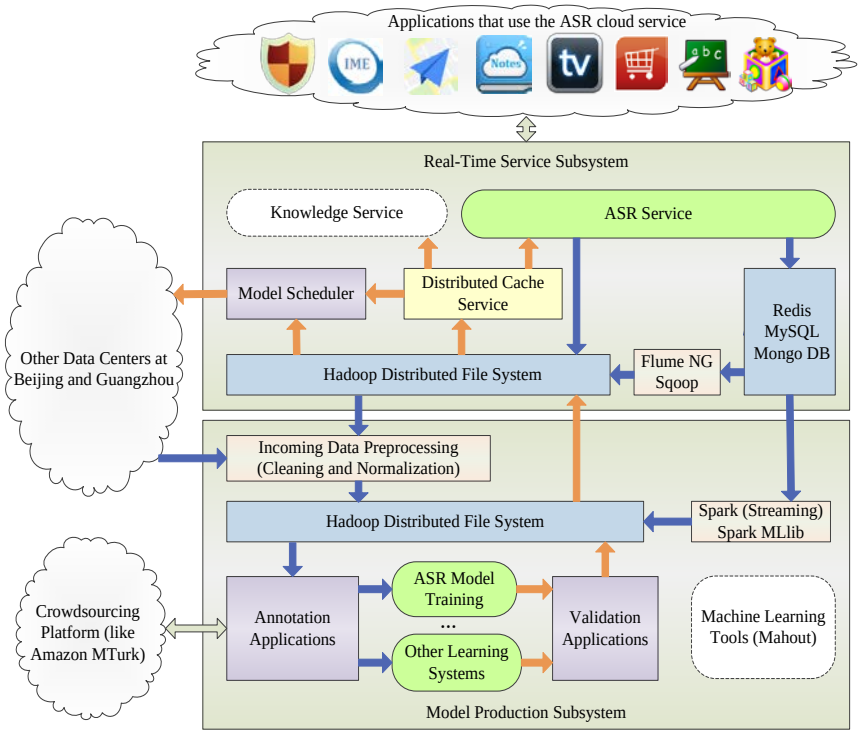


Fig. 2. Architecture of the proposed system that deliveries both real-time ASR service and data (knowledge) service. The system incorporates several state-of-the-art techniques such as crowdsourcing and lightning-fast distributed cluster computing.

4. System Architecture

The overview of the architecture of our proposed system is illustrated in **Fig. 2**. After several year developments, the system currently provides large-scale real-time ASR service for standard Chinese Mandarin language, which is adopted by a large number of customers including government institutes, industrial companies, and end users. For example, public security department uses our ASR service to search the sensitive words from the speech voices of those who are thought to be potential threats. Automobile navigator manufacturers utilise our ASR service to provide a voice control in their navigation devices. A great deal of personal mobile applications such Input Method Editor for Chinese characters, Siri-like personal assistant, electronic reader, smart TV, language trainer, and even toys for babies, are equipped with our ASR functions. All the data collected from these applications in different domains form a gold ore for us to further mine.

To provide high quality large-scale real-time and value-adding services, we divide the whole system into two logical subsystems: a Real-time Service subsystem (RS) and an offline Model Production (MP) subsystem. Notice that we already have three data centres in the different cities, the RS subsystems are installed in all data centres and the MP subsystem is only installed in the data centre equipped with the most powerful hardware, which serves as a

Main Data Center (MDC). For an RS subsystem, the user-specific model data are utilised in a read-only manner. They must be quickly retrieved and loaded in each data centre. For the MP subsystem, the data of user models are updated after offline (re-)training. To keep data consistency when producing the user-specific models, it would be better to adopt a centralized processing paradigm using only one data centre. As **Fig. 2** demonstrates, both RS and MP subsystems are built on the top of the latest release of Hadoop Distributed File System (HDFS) [3], which offers a huge scalable storage space for continuing inflows of users' generated data (voice, logs and others) and potential opportunities to analyze big data through MapReduce [5] and other machine learning tools for big data such as Mahout [4]. The incoming data (blue arrows in **Fig. 2**) in an RS system may stream into different sinks. Besides the voices of the users that may be directly stored in HDFS, logs and other data may be stored in a traditional database such as MySQL or a NoSQL database such as Redis and MongoDB. Some information that reflects the health of an RS system is extracted immediately from the logs by Flume NG [15], Sqoop [7] and HiveQL [9]. The health-related information of the system components can be received by monitors, and then the supervision tools can adjust the configuration and tune the running status of the system if some unusual events have happened.

The MP subsystem has a loosely coupled architecture. The user-generated data other than voices stored in Redis, MySQL, or MongoDB are business related. These data can be analysed and mined for a future usage (e.g., used in advertisement recommendation) by a lightning-fast distributed cluster computing engine Apache Spark (Streaming) and its machine learning library SparkMLlib. For example, we have implemented a user categorisation framework which periodically calculates the user topic models by clustering profiles of online users using the Latent Dirichlet Allocation model [2] and the K-Means algorithm implemented in SparkMLlib. This framework helps us precisely advertise to target customers. The incoming voice data from numerous mobile applications flow into an *Incoming Data Preprocessing* system, where voices from the same user (based on user's registered ID) are merged, and their corresponding transferred texts are also processed by several basic information retrieval techniques such as stemming and stop-word removal. After the data have been cleaned and normalised, they must be labelled for the subsequent supervised learning. Both kinds of personalised services require labelling data. For a personalised ASR service, we need to label the phonemes for an acoustic model and phrases for an *n-gram* model. For personalised knowledge sharing, we may need to label categorical attributes or concepts according to their specific application domains. All these annotation tasks usually require a lot of human intelligence and labours. Compared with traditional labelling by domain experts, we introduce a novel annotation scheme: labelling by crowdsourced non-expert annotators. We post data to be labelled on crowdsourcing systems, such as Amazon Mechanical Turk (MTurk), and then let multiple annotators from Internet label the same datum. The labelling accuracy has shown to be improved through this repeated labelling scheme if good consensus algorithms are carefully chosen [14]. However, due to the tendency that annotators from Internet exhibit systematical labeling biases, we have proposed several novel consensus algorithms to handle the biased labeling issue for both binary [20] and multi-class categorization problems [19], which definitely improve the performance of our annotation applications to the level of being applied to a real-world usage. Using crowdsourcing significantly reduces the cost of data labelling, which makes a personalised service for millions of people become real. The user-specific models are trained via different machine learning systems. Currently, we have

Table 3. Comparisons of Classification Accuracy in Two Tasks

Algorithms	Sentiment	Experienced
QTL	**97.52 ± 0.02**	**75.29 ± 0.02**
LR	65.57 ± 0.00	70.23 ± 0.00
DTL	82.63 ± 0.04	72.03 ± 0.03
Tri-TL	93.75 ± 0.02	58.25 ± 0.07
HIDC	92.18 ± 0.02	70.98 ± 0.02

developed a sophisticated ASR-related learning system. We are planning to deeply mine the knowledge hidden in the transferred texts. The validation programs test the newly created models and then put them into an RS subsystem if the performance of these learned models satisfies the requirements.

The RS subsystem does not include any training systems, but only uses the learned models (orange arrows in **Fig. 2**) to provide services. The most significant challenge here is to reduce the time consumption of retrieving and loading user-specific model data. For a specific user, the state-of-the-art ASR technique usually requires this user's acoustic and language model to adjust the results calculated against the common standard model. A user-specific model usually consumes several megabytes to a hundred megabytes of memory. Since the low performance of small file storage is a widely known weakness of HDFS, we have proposed a distributed cache system HDCache [21] that accelerates the speed of model retrieval and loading. HDCache is a distributed layered cache system built on the top of HDFS. The cache system consists of a client library and multiple cache services. The models are cached in the shared memory which can be directly accessed by an ASR service with a cache client library integrated. Cache services are organised in the peer-to-peer style using a distributed hash table [6]. Every cached model has three replicas in the memories of different cache services, which improves robustness and alleviates the workload. When a mobile user travels from one place to another and results in switching the data centre, the Model Scheduler in the data centre to which the user switches will immediately query the HDCache for the existence of corresponding models. If the models cannot be found in the cache or HDFS systems, this Model Scheduler will contact the Model Scheduler in MDC, fetch the required models and load them into the HDCache system in the same data centre. All these actions should be done before the user accesses the ASR service. With the distributed cache system, the average model loading time can be reduced to within 30 milliseconds [21]. HDCache was originally designed for ASR service, but it is being extended for knowledge service. Since the whole system is so complicated that some monitor, supervision, configuration, coordination, deployment tools are also included in the system for its deployment and management, which is not presented here for the sake of simplicity.

5. Experiments

Having a QTL algorithm and the real-time cloud system, we go back to our application scenarios. Currently, our system provides two prediction tasks for the public as knowledge services based on the collected data in the smart TV domain: predicting the sentiments (positive or negative) of users and predicting whether a user is an experienced one. Since the

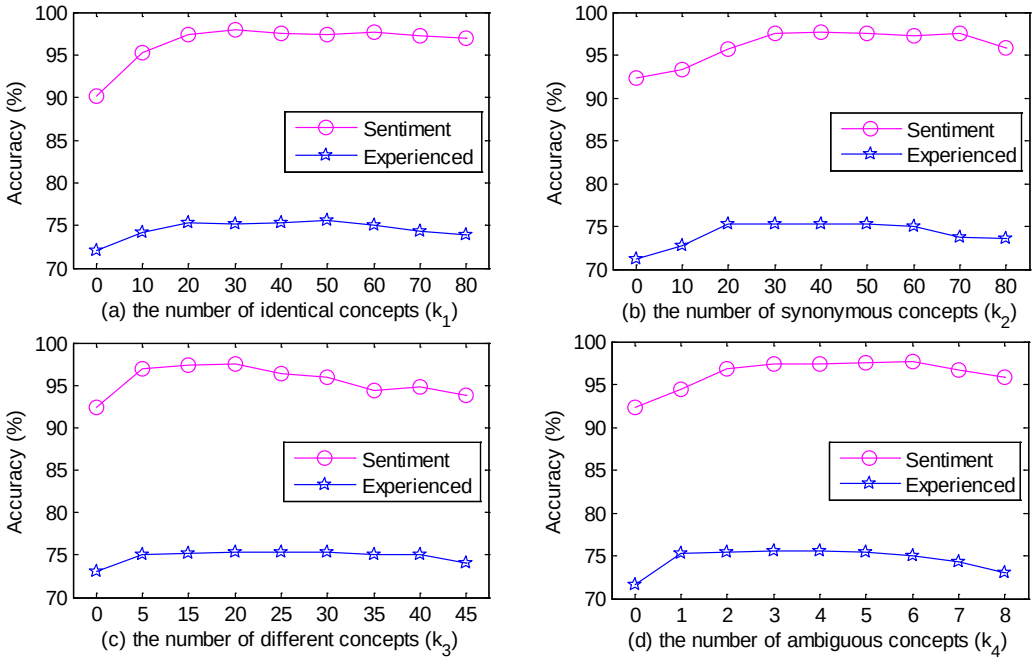


Fig. 3. Parameter sensitivity of QTL on the Sentiment and Experienced tasks.

collected data are reviews and comments to TV serials and movies that reflect the sentiments of users, we can use our QTL algorithm to build a model to predict users' sentiments towards a book in E-Books application. Similarly, users are experienced ones if they can provide profound reviews, which should be given more attention. We compared QTL with a regular learning algorithm logistic regression (LR), dual transfer learning (DTL) [10], triplex transfer learning with exploiting both shared and distinct concepts for text classification (Tri-TL) [22], and concept learning for cross-domain text classification (HIDC) [23].

In our first experiment, we set $k_1 = 30$, $k_2 = 40$, $k_3 = 20$, and $k_4 = 5$. The maximum number of iterations is set to 100. We use the accuracy as our evaluation metric. The comparison results are listed in **Table 3**. Obviously, compared with the state-of-the-art algorithms, QTL obtain the best performance on the two datasets. On *sentiment*, QTL is 3.77 points greater than the runner-up Tri-TL. On *Experienced*, OTL is 3.26 points greater than the runner-up DTL. Considering the small standard deviations in the results, the superiority of QTL is significant.

Our second experiment is to investigate the impact of the parameters k_1, k_2, k_3, and k_4 on the performance of the learned models. We increase the value of k_1 from 0 to 80 with a step size 10 and fix the values of k_2, k_3, and k_4 the same as they are in the first experiment, increase the value of k_2 from 0 to 80 with a step size 10 and fix the values of k_1, k_3, and k_4, increase the value of k_3 from 0 to 45 with a step size 5 and fix the values of k_1, k_2, and k_4, and increase the value of k_4 from 0 to 8 with a step size 1 and fix the values of k_1, k_2, and k_3. **Fig. 3** shows the parameter sensitivity of QTL on the *Sentiment* and *Experienced* tasks. Overall, the performance of QTL is not very sensitive to the values of these four parameters. For each parameter, we can easily find a relatively wide range within which the performance

of the models can maintain at a high level. The robustness of QTL makes it easy to be used in real-world scenarios.

After we train these transfer learning models in the smart TV domain, new applications in other domains, such as E-Book sales and rent, do not need to collect substantial user-generated data and label them for a learning model training. That is a value-adding service that our system provides for those who are willing to integrate our ASR function in their applications.

6. Conclusion

This paper proposed a novel ongoing service-oriented prototype system, which provides personalised services to massive mobile users. The system implements the platform as a service (PaaS), which provides an automatic speech recognition (ASR) services in a real-time paradigm. To provide a more personalised service, we connect the real-time service subsystem and the model production subsystem through the Hadoop distributed system and introduce the crowdsourced annotators to accelerate the labelling process. Through this personalised ASR services, we can efficiently collect a lot of users' voice data, and their corresponding transferred text data. Furthermore, we attempted to discover the knowledge hidden behind using machine learning techniques. The paper demonstrates a Quadruple Transfer Learning method that can take advantage of the knowledge discovered from the smart TV domain to classify user interests in an E-Book mobile application. In the future, providing a cross-domain shared knowledge platform and involving more learning algorithms are the long-run direction of the evolution of our system.

Acknowledgment

This research has been supported by the National Natural Science Foundation of China under grants 91846104, 61603186, 61703187, 61972202, 61602243, 61501241 and 61702355, the Natural Science Foundation of Jiangsu Province, China, under grants BK20160843, the China Postdoctoral Science Foundation under grant 2017T100370, and the Fundamental Research Funds for the Central Universities (30919011235, 30920120180101, and 30919011231). We especially thank the software developing and test engineers Licong Deng, Gaoyan Fan, Yan Xu, and Jin Xu for their hard working on our prototype system.

References

[1] L. Besacier, E. Barnard, A. Karpov and T. Schultz, "Automatic speech recognition for under-resourced languages: A survey," *Speech Communication*, vol. 56, pp. 85-100, 2014. Article (CrossRef Link).

[2] D. M. Blei, A. Y. Ng and M. I. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993-1022, 2003.

[3] G. Chen, H. V. Jagadish, D. Jiang, D. Maier, B. C. Ooi, K. L. Tan and W. C. Tan, "Federation in cloud data management: Challenges and opportunities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 7, pp. 1670-1678, 2014. Article (CrossRef Link).

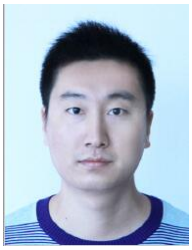
[4] T. Condie, P. Mineiro, N. Polyzotis and M. Weimer, "Machine learning for big data," in *Proc. of the 2013 ACM SIGMOD International Conference on Management of Data*, pp. 939-942, 2013. Article (CrossRef Link).

[5] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107-113, 2018. Article (CrossRef Link).

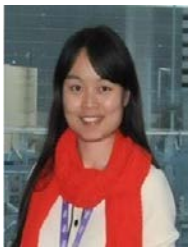
[6] W. Galuba and S. Girdzijauskas, "Distributed hash table," *Encyclopedia of Database Systems*, pp. 903-904, 2009.
[7] S. Y. Ho and S. H. Kwok, "The attraction of personalized service for users in mobile commerce: an empirical study," *ACM SIGecom Exchanges*, vol. 3, no. 4, pp. 10-18, 2002. Article (CrossRef Link).
[8] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Machine Learning*, vol. 42, no. 1-2, pp. 177-196, 2001. Article (CrossRef Link).
[9] O. V. Joldzic and D. R. Vukovic, "The impact of cluster characteristics on HiveQL query optimization," in *Proc. of the 21st IEEE Telecommunications Forum*, pp. 837-840, 2013. Article (CrossRef Link).
[10] M. Long, J. Wang, G. Ding, W. Cheng, X. Zhang and W. Wang, "Dual Transfer Learning," in *Proc. of the 2012 SIAM International Conference on Data Mining*, pp. 540-551, 2012. Article (CrossRef Link).
[11] M. Mehrabani, S. Bangalore and B. Stern, "Personalized speech recognition for Internet of Things," in *Proc. of the 2nd IEEE World Forum on Internet of Things*, pp. 369-374, 2015. Article (CrossRef Link).
[12] J. Pan, X. Hu, Y. Zhang, P. Li, Y. Lin, H. Li, W. He and L. Li, "Quadruple Transfer Learning: Exploiting both shared and non-shared concepts for text classification," *Knowledge-Based Systems*, vol. 90, pp. 199-210, 2015. Article (CrossRef Link).
[13] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359, 2010. Article (CrossRef Link).
[14] A. Sheshadri and M. Lease, "SQUARE: A benchmark for research on computing crowd consensus," in *Proc. of the First AAAI Conference on Human Computation and Crowdsourcing*, pp. 156-164, 2013.
[15] C. Wang, I. A. Rayan and K. Schwan, "Faster, larger, easier: reining real-time big data processing in cloud," in *Proc. of the Posters and Demo Track at Middleware '12*, pp. 1-2, 2012. Article (CrossRef Link).
[16] X. Wu, X. Zhu, G. Q. Wu and W Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97-107, 2014. Article (CrossRef Link).
[17] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu and G. Zweig, "The Microsoft 2016 conversational speech recognition system," in *Proc. of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5255-5259, 2017. Article (CrossRef Link).
[18] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto, C. Yu, W. J. Fabian, M. Espi, T. Higuchi and S. Araki, "The NTT CHiME-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices," in *Proc. of the 2015 IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 436-443, 2015. Article (CrossRef Link).
[19] J. Zhang, V. S. Sheng, J. Wu and X. Wu, "Multi-Class Ground Truth Inference in Crowdsourcing with Clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 4, pp. 1080-1085, 2016. Article (CrossRef Link).
[20] J. Zhang, X. Wu and V. S Sheng, "Imbalanced multiple noisy labeling," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 2, pp. 489-503, 2015. Article (CrossRef Link).
[21] J. Zhang, G. Wu, X. Hu and X. Wu, "A distributed cache for Hadoop distributed file system in real-time cloud services," in *Proc. of the 2012 ACM/IEEE 13th International Conference on Grid Computing*, pp. 12-21, 2012. Article (CrossRef Link).
[22] F. Zhuang, P. Luo, C. Du, Q. He, Z. Shi and H. Xiong, "Triplex transfer learning: exploiting both shared and distinct concepts for text classification," *IEEE Transactions on Cybernetics*, vol. 44, no. 7, pp. 1191-1203, 2014. Article (CrossRef Link).
[23] F. Zhuang, P. Luo, P. Yin, Q. He and Z. Shi, "Concept learning for cross-domain text classification: A general probabilistic framework," in *Proc. of the 2013 International Joint Conferences on Artificial Intelligence*, pp. 1960-1966, 2013.

Jing Zhang is currently an Associate Professor in the School of Computer Science and Engineering at the Nanjing University of Science and Technology. He received his M.S. degree in Computer Science from the Graduate University of Chinese Academy of Sciences, Beijing, China, in 2006, and his Ph.D. degree in Computer Science from the Hefei University of Technology, Hefei, China, in 2015. His research interests include data mining, machine learning, and their applications in business and industry. He has published dozens of articles in some prestigious international journals and conferences.

Jianhan Pan is currently an Associate Professor in the School of Computer Science and Technology, Jiangsu Normal University. He received his B.Eng. degree from Nantong University, Nantong, China, in 2007, his M.Sc. degree in Computer Science from University of Electronic Science and Technology of China, Chengdu, China, in 2012, and his Ph.D. degree from School of Computer and Information, Hefei University of Technology, China, in 2016. He His research interests include data mining and machine learning, focusing on transfer learning. He has published several articles in some prestigious journals.



Zhicheng Cai is currently an Associate Professor in the School of Computer Science and Engineering at the Nanjing University of Science and Technology. He received his B.Sc. and Ph.D. degrees in Computer Science and Engineering from Southeast University, Nanjing, China, in 2009 and 2015 respectively. His research interests focus on load prediction, dynamic capacity management and task scheduling in cloud computing. He is the author of more than 10 publications in journals such as IEEE Transactions on Services Computing, IEEE Transactions on Automation Science and Engineering, IEEE Transactions on Cloud Computing, Future Generation Computer Systems, Journal of Grid Computing, and at international conferences such as ICSOC, ICPADS, ISPA, ICA3PP, SMC, and CASE.



Min Li is currently an Associate Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. She received the Ph.D. degree in biomedical engineering from Chongqing University in 2014. From 2011 to 2013, she was a Visiting Ph.D. Student with the Department of Radiation Oncology, University of Texas MD Anderson Cancer Centre, Houston, TX, USA. Her current research interests include medical image processing, analysis, and visualization.

Lin Cui is currently a professor at Intelligent Information Processing Laboratory, Suzhou University, Anhui, China. She received the Ph.D. degree from the School of Computer Science and Information Engineering, Nanjing University of Aeronautics and Astronautics (NUAA). Her research focuses on data mining, POI recommendation, and social network analysis. Since 2008, Dr. Cui has published dozens of papers in journals such as IEEE Transactions on Systems, Man, and Cybernetics: Systems, Applied soft computing, and in conferences such as International Joint Conference on Neural Networks (IJCNN) and SIAM International Conference on Data Mining (SDM), including one SDM Best Paper Award in Applied Data Science Track.