How to Personalize Speech Applications for Web-based Information in a Car

Philipp Fischer, Andreas Österle, André Berton, Peter Regel-Brietzmann

DaimlerChrysler AG, Group Research/Telematics and Entertainment, Germany
philipp.fischer@daimlerchrysler.com

Abstract

We present a system for exploring and personalizing internet information in the car using natural language queries. Speech dialog applications are generated automatically from well-structured internet content, such as tables, and transferred to the car. In order to cope with the large variety of speech applications, we propose a hybrid content-based personalization approach. Speech applications are clustered into various topic areas by mapping them to a domain ontology. Applications are ranked according to explicit preferences of the driver, global profile data and an implicit user profile. This profile adapts itself while the user is interacting with the system and takes into account the selected application and the speech queries. The resulting list of ranked applications is displayed to the user. Global ratings are based on the preferred topics of 20 subjects that were also questioned about the prototype’s usability and accuracy in a first user evaluation.

Index Terms: semantically enhanced personalization, adaptive speech systems, intelligent user interfaces, multi-modal dialogs, in-car speech dialogs.

1. Introduction

Drivers of premium cars face an increasing number of features and choices to be made in telematics and entertainment systems. The driver must cope with this growing complexity in addition to the variety of switches and displays required for driving. It is therefore crucial that driver distraction be reduced to a minimum. This can be achieved by providing the driver with the most relevant information in a given context using an intelligent, multimodal, and adaptive interface system.

Many situations can be thought of where the driver would like to have additional information beyond that already available in cars. Drivers might be interested in finding restaurants or movie theaters nearby, free parking at the destination, cheap petrol stations, or where radar speed controls are along his way. At home most information can easily be found by browsing the internet, but standard user interaction of browsing is too complex and distracting to be done while driving. This is due to small sizes and unstructured content on small displays, which makes it hard to understand the information. When drivers have access to the internet, the preferred modality of interaction is speech since this minimizes driver distraction and reduces the distracting glances at the display.

We propose a method of personalized information filtering which ranks items (i.e. speech applications) preferred by the driver higher than other items thus minimizing his need to scroll in large lists. This in turn reduces the effort required to select items for further consideration. In our approach speech is the primary mode of interaction for accessing the internet. Information is presented by a list of headlines, one headline for each application. After starting an application manually or uttering a headline, the content of the application is presented as speech output which can be queried with spoken questions.

The system replies by reading out all relevant information to the driver.

Information filtering, personalization, and adaptivity of user interfaces are issues in many areas of application. This paper focuses on personalizing speech access to driver applications. We understand personalization as the adaptive process of filtering and recombinig information based on the user’s context and preferences that change over time. The system presents new information that might be of interest to the user by inferring the user’s preferences on unseen items from those he has looked at in the past.

1.1. Related Work

There has been a lot of work on personalization in the areas of web personalization, product recommendation systems, and adaptive user interfaces. A broad overview of web personalization techniques, including a classification into categories, is described in [1]. We use a hybrid approach in which global preferences, explicit user preferences, and implicit ratings are combined into one personalized preference rating. The implicit rating is based on information related to the content that is presented to the user (content-based personalization). Explicit profiling can improve performance if the driver creates and updates his own profile, but hybrid approaches or even carefully constructed implicit profiles can perform similar to explicit personalization [8].

Many approaches to web personalization can be applied to in-car speech-based personalization of web information. The nature and quantity of the interaction data collected from speech queries must be particularly well focused, since this data differs from web-based server data (page views, clicks, visit times). Content-based personalization approaches have become again a focus of research because of the rising interest in ontologies and the semantic web. Classical rating-based approaches can be used for ontologies by adapting the vector space model to ontology classes [2,3]. In this paper we focus on content-based personalization only. We are planning to evaluate collaborative-filtering approaches for mobile applications because they are widely used in on-line scenarios, too.

Adaptive personalized interfaces are known to improve user experience. It is imperative, however, that systems be designed so that the user understands how personalization works, in order to build up his trust. Organized interfaces [4] provide a good method for handling large sets of recommendations.

The following section describes the scenario and the application. Section 3 presents the architecture of the prototype. Section 4 then describes in detail how our personalization method is realized. The last section concludes with an outlook on future work.

2. Scenario and Application

How is a driver informed about new speech applications from the internet? The first part of the display containing standard
applications such as audio, telephone, video, and navigation is augmented by an application group called www-Info (for internet information, see Figure 1). Although all applications can be accessed with speech commands, internet information is better modeled with typical question answering, which requires natural language dialog modeling. This means the driver can ask questions as whole sentences without having to look at the display and receives replies from the system's speech synthesis module.

After starting the speech dialog by pressing Push-To-Activate, the driver requests internet information by saying “internet”. He will then be presented with a list of speech applications listed by subject, e.g. “gas prices”, “city information”, “press review”. The list on the screen is personalized and contains two parts. Question answering for a specific application can be activated by uttering the subject or number of the item. The system acknowledges the subject just selected and contains two parts. Question answering for a specific application can be activated by uttering the subject or number of the item. The system acknowledges the subject just selected by repeating the subject and including a short description of its content. Now the driver can start asking questions related to the topic area, e.g. “Where can I find the cheapest gas stations around here?”. “Give me news on politics”, or “What’s going on in Berlin?” The system analyzes the input sentence and then reads out all relevant information. The driver can terminate the dialogue by saying “That’s all” or by directly switching to another car application such as “radio” or “navigation.”

2.1. The Need for Personalization

The list of applications is first presented to the driver visually since a complete oral read-out might be boring. He must therefore still look at the display to choose the application he wants. Due to matters of usability, however, the display can hold only about 5 items depending on their size whereas a complete list of applications can easily scale to hundreds of items. A personalization method is needed to present the most important, favorite and recent items on the upper list positions and thus on the first pages to minimize scrolling.

2.2. Interface Design

In order to achieve the user’s acceptance and appreciation, personalization has to be explained to the user by organizing and structuring the interface. Hiding information or changing the user interface with no visual reason will confuse the user if he is not familiar with the system and does not want to spend time in understanding personalization abilities of the system. If the user does not understand the system, he will neither trust its answers nor accept its usefulness. In that case the user would most likely disable personalization or even stop using the system soon.

We structure information into two categories similar to organized interfaces [4]. That is done by dividing the list into “Favorite” and “New” applications. The order in both categories depends on the ranking algorithm. Headlines in the “Favorite” section remain unchanged for a longer time since they are of high interest to the user. The “New” section of the display contains new information that might be of interest, so its contents change more frequently. Up to two related topic labels are appended to the right of each item. The process of reordering items is visualized by having the buttons move to their new positions. Buttons being removed from the screen are faded out and new buttons are faded in. The user may disable and freeze personalization to help increasing the user’s trust.

3. Prototype Architecture

Our prototype generates speech dialog applications automatically from structured internet information (in the form of tables) [5]. These applications are generated off-board because of the limited computing power available on the in-car computer. A dialog application consists of the original table it is based on, various prompts and grammars, and an XML dialog description. Hundreds of speech applications can be potentially generated by searching the internet for information matching specific topics of interest to the majority of drivers.

After generating an application, the server then transfers it to several digital broadcast stations. These then package the applications and transfer them to the local car receivers. We decided to use Digital Media Broadcast (DMB) technology since modern cars come with Digital Audio Receivers that can decode DMB.

How does this new approach affect the existing in-car dialog system architecture? The common architecture of Speech Dialog Systems (SDS) has been published many times [6]. The in-car SDS typically runs on the head unit and is in charge of processing the classical car applications, such as phone, navigation, and audio. Figure 2 shows a typical SDS architecture, which consists of standard components for Automatic Speech Recognition (ASR), Text-To-Speech Synthesis (TTS), and a Command-and-Control Dialog Manager (C&C DM) which can process fixed speech commands. The SDS also contains a module to synchronize speech with haptic input and graphical output. Synchronization is initiated by various events and dynamic data.

The Graphical User Interface (GUI) follows the model-view-controller paradigm. The interaction is modeled by state charts and events; views are defined by widgets; and a controller handles events on input/output devices, a central control switch, and a touch-screen display. This architecture must be augmented in order to make internet information accessible by voice in a car. To this end a new dialog manager for processing question answering (NatLang DM) [7] was added. It contains a natural language understanding component, not yet available in the C&C DM. The Meta Dialog Manager (MDM) controls which of the two dialog managers receives a given activity. This extension also includes the Filter-&-Update module, which accepts new speech applications from the DMB receiver, recompiles the dialog model for the NatLang DM, and the vocabulary-and-language model for ASR, where the phonetic transcription for any out-of-
vocabulary word is generated by the grapheme-to-phoneme converter of the TTS module.

The Filter-&-Update module not only updates the NatLang DM and ASR, but also sends the names and details on the new speech applications to the C&C DM and to the GUI via the MDM, so that the names and details can be shown on the display and accessed by voice and the central control switch.

The personalization module, which is new in the system, calculates new rankings whenever an interface method is called by MDM after user interaction. It is notified by the Filter-&-Update module about new available applications. Update strategies for internal ratings are assigned to specific access types while initializing the component. Ontology classes are retrieved from an OWL (Ontology Web Language) file. In order to be referenced for storing rating values, items and ontology classes have to be prepared in a personalization data model. All registered update strategies for internal ratings are executed while the user is interacting with the system. If the order of personalized items has changed, the GUI module is asked to update the list of headlines to reflect the current personalization state.

4. Personalization Method

Personalization is performed by a client-side component in the car. We decided to use semantically enhanced content-based personalization, since ontology processing may produce better ratings. Furthermore, privacy issues for the driver could arise if collaborative approaches need to upload data from the car to a server component.

Personalization can be conceptually thought of as a function that estimates user preferences. When applied to a new speech application, it returns a score which reflects its importance to the user. The parameters of this utility function are optimized by data collected from user interaction. This is done by storing rating values for each item to be personalized in a user profile. User preferences are estimated by updating and evaluating all rating values according to a specific personalization method. Our personalization method computes an overall rating for each speech application based on three rating schemes: an explicit user profile, implicit data based on user interaction, and a global rating profile. We assume that combining these ratings will lead to an order of applications which reflects user preference. Ratings may differ depending on the way they are created and updated. By combining the ratings from different sources, it becomes possible to up the rating of an item which is missing one individual rating.

4.1. System Ontology

In our case, content-based personalization means matching speech applications automatically to one or more classes of an ontology. The membership value of an speech application to an ontology class is computed from the frequency of words in its topic heading and associated description.

We built a domain ontology using OWL (Figure 3) which consists of a hierarchical topic class with a depth of 4 levels and a location class. We use a topic hierarchy with 9 main topics and a total of 115 topics in a tree with up to 4 levels. This ontology is used to match all speech applications to topics by creating instances of speech applications.

We are also considering incorporating temporal information to prioritize the list of headlines by time of day. Temporal information can be included in an ontology as a contextual class, the same way locations are now handled.

4.2. Explicit and Implicit Personalization

Explicit preferences can be selected by the user on a screen with 9 topics representing the root topics of the ontology. Explicit profiles provide direct information on the user’s preferences, but they may not be updated by the user or maybe not even be initialized. Profiles solely based on implicit data have to deal with initial lack of rating data.

Implicit preferences are based on the speech applications the user has interacted with and the wording he used for input. To infer a user’s preferences, ratings can be stored for each ontology class and for each instance of the speech application class.

Currently 3 strategies for updating ratings are implemented. The explicit strategy sets ratings to a given value and maps the user’s settings into system values. The second strategy updates ratings gradually based on access frequency. The rating of an item is increased on successful recognition whenever the user explores an item by speech or whenever an item is chosen from a list by haptic input. The propagating strategy rates other OWL classes connected to the class on an item instance. We handle subclassOf relations and OWL property relations as connections. Nodes up to 3 links away from the origin node get rated and multiplied by weights that are uniformly decreased with increasing distance from the origin. The degree of membership, i.e. to what degree a speech application is matched to a particular topic, also influences the calculation. All update strategies scale the ratings and can be configured to take into account a number of ratings in history.

Figure 2: Speech dialog system architecture

Figure 3: Parts of the domain ontology
4.3. Structure of a User Profile

The overall rating for a given topic is the weighted sum of five individual ratings as shown in Table 1. This overall rating is used for prioritizing a list of items according to the user’s preferences. More details will be described in the following sections.

<table>
<thead>
<tr>
<th>Items ratings</th>
<th>Implicit ratings</th>
<th>Topic ratings</th>
<th>Location ratings</th>
<th>Explicit ratings</th>
<th>Global ratings</th>
</tr>
</thead>
<tbody>
<tr>
<td>5%</td>
<td>25%</td>
<td>15%</td>
<td>45%</td>
<td>10%</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Rating weights

4.3.1. Implicit ratings

Dynamic profile information consists of ratings for ontology classes, and ratings for actual topics representing real word instances of an ontology class, i.e. the headlines shown on the screen. Whenever the user interacts with the system, his ratings are updated by every update strategy that is associated with this type of interaction.

For frequency-based access to topic classes, location classes and items are counted and normalized. Ontology knowledge for topic- and location-based ratings is included in addition to ratings for the item itself. We create pairs of implicit ratings $R_{ij}$ for each item $I_j, 1 \leq i \leq m$ by including the ratings of each relating concept class $C_j, 1 \leq j \leq n$ according to the relative membership value $m(I_j, C_j)$ A value $p$ is used to scale and normalize ratings.

$$R_i = \frac{1}{p} \sum_{j=1}^{n} R(C_j) \times m(I_j, C_j)$$ (1)

4.3.2. Explicit ratings

Explicit ratings for an ontology’s root topics are acquired from the driver’s settings. The system is able to store gradual positive ratings for each topic, but the user interface allows only enabling or disabling topics in the current state of implementation. Explicit ratings are assigned a weight of 0.45 for the overall rating.

4.3.3. Global ratings

For simulating a collaborative scenario a global profile consisting of the favorite topics was created by a user evaluation with 20 subjects (described in section 4.4). Ratings for building the global profile were set to the relative value of how often topics were accessed in the user evaluation. The rating values are: traveling (0.43), car (0.15), living (0.14), entertainment (0.08), sports (0.06), economics (0.04), health (0.03), computers (0.02). Global ratings are weighted by 0.1 for the overall user profile.

4.4. User Evaluation

During a user evaluation with 20 subjects, a proof-of-concept using 50 speech applications for our personalization approach was delivered. Subjects were asked to successively access speech applications according to their own personal choice while pretending to be driving a car. The average length of sequences of accessed speech application was 8.9. We asked the subjects for their personal feedback on system accuracy and system design. The list position of each speech application was measured whenever an item had to be searched for the next access based on the sequence in the given scenario. Our evaluation criterion is the overall average position of the item in question which was 17.25. We think the system could adapt itself better to a given driver if the driving sessions were longer and more than one session was used. This needs to be further evaluated.

5. Conclusions and Perspectives

In this paper we presented the system architecture for a speech-based system for accessing web information in a car. We focused on the personalization method for ranking speech applications according to their relevance to the driver. Our method takes into account explicit user preferences, recent speech interaction, and a global profile. The user profile is adapted by a hybrid content-based approach of implicit and explicit ratings. We defined an ontology model to match speech applications to topics and locations thereby allowing items to be personalized by their semantic context. Our user evaluation showed a general acceptance of our system and has helped us create initial global profile ratings for each topic in the ontology.

In a next step we will enhance personalization with temporal information, since the temporal context might help users get the most relevant information at a particular time, such as during rush-hour or at dinner time. Due to privacy concerns of the driver, we must critically evaluate whether collaborative filtering techniques can or should be added, even if they are optional.

6. Acknowledgements

This work was partially funded by the German Ministry of Education and Research BMBF as part of the SmartWeb project. We would like to thank Dr. Sunil Thakar and David Stall for their valuable comments.

7. References