INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Context-Aware Restaurant Recommendation for Natural Language Queries: A Formative User Study in the Automotive Domain

Philipp Fischer[1], *Cornelius Styp von Rekowski*[2], *Andreas Nürnberger*[2]

[1]Mercedes-Benz R&D North America Inc., CA, USA
[2]Otto-von-Guericke-University Magdeburg, Germany

philipp.fischer@daimler.com, cornelius.styp@st.ovgu.de, andreas.nuernberger@ovgu.de

Abstract

In this paper, the authors describe an extension to an approach previously discussed for personalization of a natural language system in the automotive domain that allows reasoning under uncertainty with incomplete preference structures. Therefore, the concept of an "information stream" is defined as an underlying model for real-time recommendation learned from previous speech queries. The stream captures contextual data based on implicit feedback from the user's speech utterances.

Furthermore, a formative user study is discussed. Each study iteration has been based on a prototype that allows the user to utter natural language queries in the restaurant domain. The system responds with a ranked list of restaurant recommendations in relation to the user's context. Several driving scenarios with varying contexts have been analyzed (e.g. weekday/weekend, route destinations, traffic). Users could inspect the result lists and indicate the most preferred item. In addition to quantitative data gained from this interaction, feedback on relevance of context features and on the UI concept was collected in a post-study interview for each iteration. Based on the study findings, we outline the contextual features found to be most relevant for speech-based interaction in automotive applications. These findings will be integrated into an existing hybrid recommendation model.

Index Terms: context-awareness, restaurant recommendation, multimodal interaction in automotive domain, applications for natural language processing

1. Introduction

Most of the more recent real world applications for speech-based conversational systems can be found running on mobile devices (e.g. Siri, Google Now, Cortana). But what is special about applying such systems in the automotive domain?

When designing interactions for automotive scenarios, minimizing cognitive load plays a more important role in comparison to interaction on mobile devices to avoid distraction. Some studies have shown that cognitive load can be reduced through the modality effect by adding speech as an interaction channel as discussed in [1, 2, 3] for specific situations. Another way to minimize distraction for the driver is reducing the number of items presented on the screen to approximately five.

Another important focus for research specifically in the automotive domain is the strong relation to fast changing contextual factors such as location paired with inherent navigation use cases. Building and researching computational intelligence for these use cases recently got more feasible through technical opportunities such as on-board computation systems with fairly reliable connection to large server-based processing power.

Context in these applications can be seen primarily as location, but also daytime, weekday and more higher-level contextual aspects such as home or work location, travelling for business or travelling for vacation.

The *information stream* is discussed as a conceptual framework to model an ad-hoc and context-related selection of items that will get visually assembled for interaction. The content of a stream will change in accordance to contextual changes and (expected) user preferences under context. Preferences can be based on content-related properties, for example - a user might like restaurants with italian cuisine when dining in the home area, but not when visiting foreign countries on a business trip.

The stream accompanies the user's intents and interests by storing preferences under context in a suitable data model that gets adjusted in real-time. The information model can recommend those items for interaction that are most important to the user.

2. Related Work

The basic task of a recommender system is to estimate ratings for items that a user has not seen before. A comprehensive review can be found in [4]. Three main approaches can be distinguished: Content-based recommendation approaches will use properties of the content and rank the result list based on the user's preferences for these content properties whereas in collaborative systems, user profiles of many users are compared to each other. Items preferred by one user will be recommended to other users that are similar to the current one. Hybrid recommenders combine these two approaches.

Context-aware recommender systems extend the basic $user \times item$ model to a $user \times item \times context$ model. Context can mean any relevant attribute such as for example time, location or the company of other people. In some applications (e.g. restaurant recommendation) it makes a big difference, when and who you are searching with [5].

Another important aspect is the presentation at the user interface. Trust and privacy concerns play an important role in how users perceive results and are willing to share personal data in favor of receiving recommendation. Several Methods have been suggested such as explaining the Recommendation Rationale to the user [6] and structuring the User Interface [7].

One of few studies in the automotive domain was conducted with 33 participants performing 12 tasks, for example finding restaurants or hotels using language-based queries,whereas the difficulty to achieve each task was measured using physiological sensors [8]. In [9], an evaluation about recommending gas stations is described. The remaining gas level and available price information were found to be important factors for users.

http://dx.doi.org/10.21437/Interspeech.2016-1503

3. Information Model and Framework

In this chapter, an information model supporting multi-valued logic is described that can suit the goal of recommending items to the user under uncertainty in relation to the user's preferences. Essentially, an information model is the data structure that stores the representations of real world entities, the related classes or categories and the preference structures for computational processing. Individual entities of discourse will be further noted with the term "content items" or - in short - "items".

3.1. Content Items

Most modern computational models are concerned with capturing the vagueness or uncertainty involved when processing a set of data items, whereas each item is a model (representation) of a real world entity. Any representation only captures certain aspects (features) of the true entity and omits other aspects- and therefore reasoning based on such a model is inherently uncertain.

An interesting approach that captures uncertainty involved when categorizing or clustering items into sets is the rough set theory by Zdzislaw Pawlak. Information objects (items) and their relations are characterized by means of the available information about them. Pawlak introduces the indiscernibility relation as a basis of rough set theory [10, 11]. An item is defined as a uniquely identifiable entity out of a larger set I of entities, which is informationally distinguishable to other items. Each object is defined by the values of a multidimensional vector of property values for this object. The items in a set have comparable features and share common semantics in perspective of the user.

Content items will be denoted by the specific element i_j out of the set of content items $I = \{i_j \mid 1 \leq j \leq n\}$. A subset of content items is usually presented to the user at the user interface, for example to be considered for selection in a list of items. The set of items is the base set that the preference model and context model is attached to. The items are the target for the outcome of the algorithm (ranking or structuring).

3.2. Classes and Categories

Clusters (or Categories, Classes) are the representation of similarities of information entities.

C ... universe of objects

A ... set of attributes

V_a ... set of values v of attributes a

$I = (C, A)$ ... is considered the information system based on universe of objects and set if attributes $(o, a), where\ o \in C\ and\ a \in A$.

The binary relation $I(A)$ on U is called an indiscernibility relation for each pair of objects o_1 and o_2, if and only if $a(o_1)$ = $a(o_2)$ for every attribute $a \in A$. $a(o_1)$ and $a(o_2)$ are specific values for the attributes. In other words, the objects are characterized by the same available information about them and are therefore considered indiscernible (similar) in view of this characerization. Elementary sets can be formed by such indiscernible objects and Pawlak often describes them as basic *granule* or *atoms of knowledge*. The interesting aspects of this approach are, that classes and categories are rather dynamic approximations emerging out of available information about each entity and enable algorithms to take into account the uncertainity of the model.

3.3. Information Streams

The term information stream (or data stream, in short *stream*) refers to a subsequent selection of available data items out of a time-variant collection of data items considered to be in the same class. In the examples, a stream will be further noted as S.

Items of a stream are those items that exist at a given moment in time in this collection. The concrete discernible items contained in the stream will vary based on the chosen point of time. The processing of the stream S_t refers to processing of the momentary data items in that collection at time t.

The processing of $P(S)$ refers to an algorithmic procedure applied to a stream S in its entireness, i.e. an algorithm that will affect the current selection of items within the stream.

The idea of such a concept is very common in time series analysis. The term *Information Stream* specifically refers to time-variant *information* of a system, whereas information has a specific definition attached.

Consider an active user who has friendly relations with 127 other users in a given information system. The recommendation task is to recommend restaurants that other friends currently have indicated to dine at (either by posting about them or checking into location). The collection of postings with restaurants from these friends would be considered the information stream about restaurants.

The *Continuity* of an information stream is a measurement for the variability of the collection of data items - in other words, how often these items will change.

3.4. Framework

The conceptual basis of the information system as described above is encapsulated in a software framework for prototyping of data-driven and preference-adaptive user interfaces and visualizations. One of the software patterns in this framework is that certain update strategies can be bound to specific input events such as spoken queries, haptical input and others. Whenever the user interacts with the system - depending on the state of the UI - the input event will not only animate the respective visual changes ("transitions") using the event-driven state engine, but also trigger the associated updates to the preference model through the registered recommender instances. Each recommender has access to the data structures of the information model as described above.

A concrete application will implement a common recommender class. The respective functions are called when the input triggers, followed by an update to preference values in the model. On display of the list, the result ratings are calculated. These base patters in "lemonflow" can suit as a starting point and building blocks in an iterative setup such as required for formative studies. The concrete recommender implementation and algorithm has been improved during each iteration of the study informed by the results from the previous iteration as described below. The research framework is released at [12] with more detailed documentation for the interested reader.

4. User Study

We performed an iterative user study following a formative evaluation approach. This particular method has been chosen to research expectations with direct feedback from the participants and integrate new suggestions directly into the Proof-of-Concept (PoC) application.

The focus of the study was on the influence of contextual aspects for recommendation in the use case of restaurant search using natural language queries. Test persons were introduced in several different hypothetical scenarios with varying context parameters. The interaction with the system has been monitored and was enhanced by a post-study interview to validate findings directly with the participants.

4.1. User Story

The study design as well as the PoC development has been aligned with a primary user story that has been defined at the beginning of the study to guide the formative evaluation:

On a Friday evening, Tom is on the way to Stuttgart to attend a small conference over the weekend. The navigation system shows the estimated arrival in 32 minutes at the hotel location. Tom's evening is entirely free and he would like to eat something, but he does not know Stuttgart very well. As he cannot use the phone while driving, he queries the internet search of his car for suggestions: "Where can I have dinner this evening?"
A result list of generally popular restaurants is displayed. By matching Tom's profile, which says that he likes to go to traditional restaurants when travelling to new cities, the system additionally shows results in a further section, for example a Swabian tavern. Tom now has a good overview on generally popular places and some that are interesting to him personally in his current context. He selects the tavern to get detailed information and makes a reservation for the evening.

4.2. Recommender System and User Interface

The application presented in this study was extended with each performed study iteration, so that user feedback from earlier stages could directly be integrated and reviewed in later iterations. Requests to the system can be made with natural language, as this is very important for the intended application in the vehicle. The textual representation of the recognized spoken request is always presented to the user. Additionally, the context of the current scenario is illustrated by images and some information such as the given day and time.

On a request, the interface graphically presents two lists of suggestions to the user: one with general, unpersonalized results and one where the entries have been adapted to the user's preferences in the respective context. Each item can be selected to see more detailed information and to finally configure it as destination for the navigation system. These user interactions with the system are recorded and used for improving the underlying recommendation model.

Our model was designed as a hybrid recommender system that mainly uses content information on previously selected items such as assigned categories, but also explicit ratings from other users and knowledge on the current user's location. These different factors are combined analogously to the approach presented in [13] with the weighting shown in table 1. The weighting was determined empirically in one step of the study by letting the users estimate how much influence each factor should have on their personal recommendations.

4.3. Study Design

The study was performed in Germany with 20 test persons, some participating in multiple iterations. This way we had participants who could assess the improvement of the system as well as unbiased people who introduced new perspectives to the study. The participants' ages were in the range from 23 to 55

Context specific		Item specific	
Item Preference	**Categories Preference**	**Explicit Rating**	**Spacial Distance**
28%	28%	23%	21%

Table 1: Influences of the different factors on the hybrid recommendation model. Preferences for discrete items and assigned categories inferred from previous selections depend on the current user context, while explicit ratings and the spacial distance only depend on the item that shall be ranked.

with 32.6 being the average age. The rate of female test persons was 60%. 40% were students, while the others were employees at Daimler, but all participants were familiar with the use of natural language systems in the car.

Three iterations were performed and the prototype has been refined with each iteration based on previous input. All study iterations were run in a desktop simulation that shows the most important aspects of a navigation system (destination, estimated arrival time, etc.). The interaction of participants with the system has been monitored and documented continuously during the study.

Two evaluation techniques were applied in each iteration:

1. Usability Tests: Users were asked to give requests to the application and to inspect the results. Their behavior (e.g. number of requests, position of selected item) as well as the system performance were observed by the test supervisor and comments were collected.
2. Interviews: After performing the given tasks, the users were asked for their rating of the system. Furthermore, their estimation on certain aspects (e.g. relevance of context features) was noted.

Different scenarios with special context information were simulated on the interface and should help users imagining their behavior in these situations. Three main scenarios were used for the tests:

Lunch Break On a weekday, the participant is making a lunch break with colleagues, so they are looking for a place to eat near their office. No destination is defined in the navigation system.

Driving Home The participant is on the way home after work on a weekday and will arrive in about 30 minutes. The home address is set as route destination.

On the way to a Conference On a Friday afternoon, the participant is on the way to a city that he/she is not familiar with. There will be a small conference over the weekend, but this evening is free. Therefore, the user is looking for a place (restaurant, bar, etc.) to go to after the arrival in about one hour.

These scenarios were chosen to potentially have an influence on the user's decision in the restaurant search use case. To make the setting as realistic as possible for each user, there was a short configuration phase at the beginning of the experiments, in which appropriate cities and times for the scenarios were set.

In the first two study iterations, the user behavior in single isolated situations was examined. The third and final iteration had the objective to simulate a longer course of interaction, so that the system could really learn the user's context specific preferences. This was achieved by going through a series of repetitive scenarios of the categories described above. The order was set up as follows:

Lunch → Home → Lunch → Conference → Home → Lunch → Conference

This sequence was not meant to simulate consecutive days in one week, but rather single events from a longer time span like several months.

4.4. Results

Several observations on the acceptance of a context-aware recommender system in the automotive domain could be drawn from the study:

- Most people inherently expect the information system to use contextual information if already known to the system. For example when a destination was configured in the navigation system, many requests were in the form *"I want to eat Italian food this evening."* without defining a particular location.
- When asked directly if they would want the system to use such data and create context-specific user profiles, the participants gave an average rating of 3.4 (on a scale from 0 to 4, 4 being best).
- More than 68% of all restaurant selections were made from the personalized result list (compared to the general results).
- Users do not like to repeat requests (on average 1.49 requests were made per situation) and tend to select items from the top of the list (65% of selected items were on positions 1 to 3).

These findings emphasize the importance of a well-performing recommender system especially in the car and confirm its appreciation by the users.

The distribution of reasons that users indicated for their selections is as follows:

categories are interesting/match the request	52.86%
good **rating**, high number of reviews	27.14%
restaurant is **known** personally	18.57%
name sounds attractive	11.43%
short **distance**	11.43%
place matches the search **location**	5.71%
restaurant offers **variety** to other selections	5.71%
restaurant has been **selected before**	2.86%
image looks attractive	1.43%

It can be seen that in restaurant recommendation it is most important to estimate appropriate preferences for assigned categories, while also the user's personal selection history needs to be regarded.

For the user interface, the following extension points were found:

- Most users mentioned to prefer one single list to choose from instead of two.
- The main reason for recommending an item should be shown.
- Unwanted entries (misconceptions of the recommendation system) need to be removable by the user.

Finally, the interview part of the experiments was used to estimate an optimal initial configuration for the parameter set of the recommendation system. Each feature could be rated on a scale from 0 to 4 with 0 meaning the feature should have no influence on recommendations and 4 meaning it should have strong influence. As can be seen in figure 1, ratings for some features such as the daytime, the frequency of visiting the destination, the distinction between weekday and weekend or the traffic on the route stand out, so these should have the biggest influence in an initial system configuration.

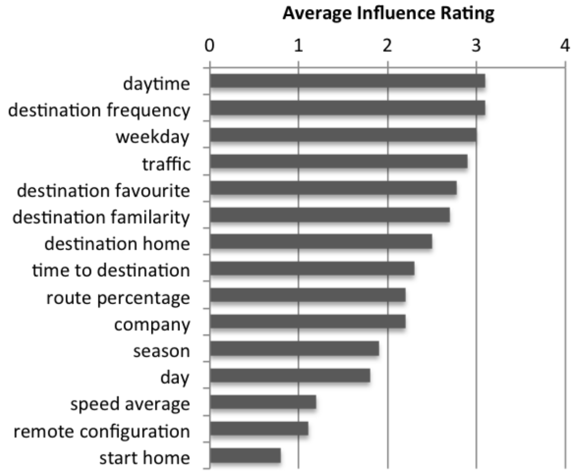

Figure 1: Interview results for the preferred influences of special context features on context-aware recommendation. A rating of 0 expresses "no influence", 4 expresses "strong influence".

5. Conclusions

The expectations of users indicate that they generally expect the system to include contextual data. If asked explicitly, users indicated an average approval of 3.4 on a scale from 0 to 4. Furthermore, the appreciation of our proposed hybrid recommender has been confirmed by the users and an initial configuration for the context module has been determined, focussing on daytime, destination frequency, weekday and current traffic.

For the interface of such an application, the following features are important: There should be only one list of results that smartly combines personalized suggestions and new ideas, the reasons for single recommendations have to be illustrated in some form and the system needs to consider both positive and negative user feedback (e.g. selections and deletions).

Our study provided first indications on how to design a context-aware recommendation system for application in the car. Now the used framework needs to be extended algorithmically, which includes finding an appropriate mathematical context representation, calculating context-specific preference values for items and categories and combining different rating aspects (e.g. category preferences, explicit ratings, distance) into one overall rating score for each item in a way that each aspect has a reasonable influence on the outcome.

Future work will include the integration of these extensions and the study findings into the framework. A summative study will then be performed with an in-car user interface based on an existing product implementation. Furthermore, with large enough data sets, state-of-the-art methods for automatically extracting important context features should be researched on this application.

6. References

[1] S. Tindall-Ford, P. Chandler, and J. Sweller, "When two sensory modes are better than one." *Journal of experimental psychology: Applied*, vol. 3, no. 4, p. 257, 1997.

[2] P. Ginns, "Meta-analysis of the modality effect," *Learning and Instruction*, vol. 15, no. 4, pp. 313–331, 2005.

[3] J. Reinwein, "Does the modality effect exist? and if so, which modality effect?" *Journal of psycholinguistic research*, vol. 41, no. 1, pp. 1–32, 2012.

[4] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, jun 2005. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2005.99

[5] ——, "Context-aware recommender systems," pp. 217–253, 2011.

[6] D. McSherry, "Explanation in recommender systems," *Artificial Intelligence Review*, vol. 24, no. 2, pp. 179–197, 2005.

[7] P. Pu, L. Chen, and R. Hu, "Evaluating recommender systems from the user's perspective: Survey of the state of the art," *User Modeling and User-Adapted Interaction*, vol. 22, no. 4-5, pp. 317–355, 2012.

[8] A. Gruenstein, J. Orszulak, S. Liu, S. Roberts, J. Zabel, B. Reimer, B. Mehler, S. Seneff, J. Glass, and J. Coughlin, "City browser: Developing a conversational automotive hmi," in *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '09. New York, NY, USA: ACM, 2009, pp. 4291–4296.

[9] R. Bader, E. Neufeld, W. Woerndl, and V. Prinz, "Context-aware poi recommendations in an automotive scenario using multi-criteria decision making methods," in *Proceedings of the 2011 Workshop on Context-awareness in Retrieval and Recommendation*. ACM, 2011, pp. 23–30.

[10] Z. Pawlak and A. Skowron, "Rudiments of rough sets," *Information sciences*, vol. 177, no. 1, pp. 3–27, 2007.

[11] Z. Pawlak, "Rough sets," *International Journal of Computer & Information Sciences*, vol. 11, no. 5, pp. 341–356, 1982.

[12] P. Fischer, "lemonflow - framework for prototyping data-driven user interaction," `http://lemonflow.github.io/` (2005-2016).

[13] P. Fischer, A. Österle, A. Berton, and P. Regel-Brietzmann, "How to personalize speech applications for web-based information in a car," in *8th Annual Conference of the International Speech Communication Association*. Red Hook, NY: Curran, 2007.