Procesamiento del Lenguaje Natural, núm. 43 (2009), pp. 77-84 recibido 1-05-2009; aceptado 5-06-2009

Managing Speaker Identity and User Profiles in a Spoken Dialogue System*

Gestión de la identidad de locutor y perfiles de usuario en un sistema de diálogo

J.M. Lucas, F. Fernández, J. Salazar, J. Ferreiros, R. San Segundo
Grupo de Tecnología del Habla, Universidad Politécnica de Madrid
Avenida Complutense s/n 28040. Madrid
juanmak@die.upm.es, efhes@die.upm.es, jsc@die.upm.es,
jfl@die.upm.es, lapiz@die.upm.es

Resumen: En este artículo presentamos una técnica para obtener preferencias de los usuarios durante sus interacciones con un sistema de diálogo hablado, aprovechando la información contenida en dichas preferencias en el módulo de gestión de diálogo. Mediante el empleo de un sistema no supervisado de identificación de locutor y un módulo de comprensión del lenguaje natural podemos adaptar el comportamiento del sistema a cada usuario particular, mediante un análisis estadístico de los objetivos preferidos por el usuario. De esta manera, podemos adaptar el comportamiento del sistema a las preferencias de cada usuario, anticipando las tareas que un usuario dado desea realizar. Hemos evaluado el sistema de identificación de locutor con dos bases de datos diferentes, obteniendo errores de identificación inferiores al 4%. La evaluación inicial del gestor de perfiles muestra una mejora de métricas subjetivas, tales como la percepción de eficiencia o la naturalidad de las respuestas del sistema.
Palabras clave: Sistemas de diálogo, perfiles de usuario, preferencias, gestión de diálogo

Abstract: In this paper we present an approach to capture user preferences during their interactions with a spoken dialogue system (SDS) and to exploit these preferences into the dialogue manager module. We use the output of an unsupervised speaker identification module and a natural language understanding module to adapt the behaviour of the SDS to each particular user by developing a statistic analysis of the preferred goals of each user. This way we can adapt the behaviour of the system to each user's preference, anticipating the goals that a user wants to fulfill. We have evaluated the speaker identification system with two different databases, and we have obtained identification errors below 4%. The initial evaluation of the profile manager shows an improvement on subjective metrics, such as users' perception of efficiency or naturalness of the system responses.
Keywords: Spoken dialogue system, user profile, user preferences, dialogue management

1 Introduction

Environmental intelligence is receiving an increasing effort in research. Developing systems that can interact with users in such a way that the user does not realize the presence of those systems is a highly attractive field. In this scope the ability of interacting with a system using speech is a must, since speech is the most natural mean of communication between humans.

There are four main situations in which user models can play a relevant role (Webb, Pazzani, and Billsus, 2001): the behaviour of different users, their degree of expertise, their preferences, and their characteristics. This work focuses on user behaviours, preferences and characteristics (if they are allowed to fulfill certain actions over the application) while interacting with a spoken dialogue system.

User modeling may be appliable to each component of a SDS (Zukerman and Lit-

* This work has been partially funded by the Spanish Ministry of Science and Innovation under contract TIN2005-08660C04-04 (EDECAN-UPM) and by UPM-DGUI-CAM under contract CCG07-UPM/TIC-1823 (ANETO).

man, 2001). For instance, the Automatic Speech Recognition module can make use of user-dependent information to adapt the acoustic and linguistic models it uses to better recognize the speech of a given user. User models can reflect the degree of expertise of each user, modifying the behaviour of the Dialogue Manager and the output of the Natural Response Generator module in such a way that it gives more information to novice users. Statistic user modeling has also been employed to improve the Dialogue Manager strategies within the scope of a reinforcement-learning approach (Schatzmann et al., 2005; Schatzmann et al., 2006). Finally, user modeling has been used to design techniques for evaluating SDS by means of simulating the behaviour of real users (Eckert, Levin, and Pieraccini, 1997).

We have implemented a first approach to add user information to a SDS which we use to control a Hi-Fi audio device using speech. To achieve this goal we have integrated a speaker identifier to our baseline system. A second task has been to develop an initial approach of a User Information Manager (UIM), which helps us to create, load, use and update user profiles. Our module has been designed taking into account the real-time constraints that both a SDS and our application domain impose.

The rest of the paper is organized as follows. Section 2 briefly presents the dialogue system which we have modified. Next, section 3 describes the main characteristics of our user information managing approach. Section 4 shows the initial experimental environment we have set up. Finally, section 5 summarizes the conclusions we have come to, and presents our future research guidelines.

2 Baseline System

The SDS we have modified (Fernández et al., 2005) consists of several interconnected modules, each of which develops a fully independent task. The information flow throughout the SDS is as follows.

First of all, an Automatic Speech Recognition module extracts the word sequence from a previously sampled speech input. Our recognizer follows an approach of Hidden Markov Models to extract the most probable sequence of words.

In a second stage the Natural Language Understanding module analyzes this word sequence to get its semantic content. We apply a rule-based understanding algorithm to associate the different possible meanings of each recognized word with a set of *concepts*, which contain the available meanings for the application domain.

The concepts extracted from each utterance are then processed by the Dialogue Manager (DM). This one applies a Bayesian Network (BN) approach to infer what objectives the user wants to fulfill, that is, the actions he/she wants to make over the final system. Each different action is associated with a *dialogue goal*. In this situation, a given utterance may have reference to several actions related to different goals.

Several natural language features, such as ellipsis or contextual references can be tackled with a Context Manager (CM). This module stores information about previous dialogue turns, which are taken into account to solve dialogues with incomplete information. That is, dialogues in which the user refers to his/her previous interactions. Using the CM our system can solve that situations without asking the user again for information he/she has provided before.

In the last stages of the dialogue flow, the DM generates a set of answer concepts, which contain the semantics of the answer it will provide to the user. The Natural Response Generator makes then use of those output concepts to build a sentence which the Text To Speech (TTS) module synthesizes.

Our approach follows the definitions of sessions and dialogue turns presented in (Pargellis, Kuo, and Lee, 2004). A *dialogue session* is a set of dialogues which starts with an activation command and finishes with a close command. Each *dialogue* contains one or more goals that the user wants to fulfill. This can be done in several *dialogue turns*, that are defined as interactions in which both the user and the system say something to each other.

Our baseline SDS (Fernández et al., 2005; Fernández et al., 2008) has been modified to include the user information manager (UIM). Figure 1 shows the complete system, as well as the interrelationships between each of its modules.

Our first modification is the inclusion of a speaker identification (SID) module which obtains the identity of the current user. The knowledge of the identity of a user will allow

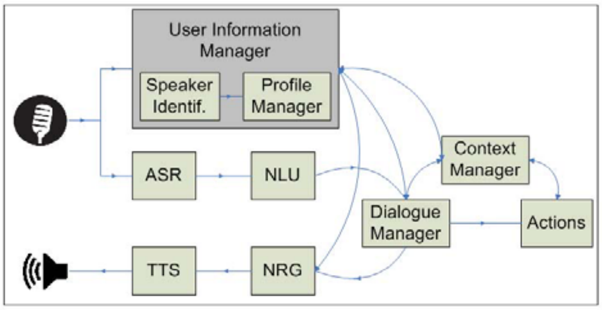

Figure 1: Block diagram of the dialogue system and the UIM

the system to build and load a data structure which contains information related to that user. The UIM will develop these tasks.

Ideally, the UIM should affect each module of the system, adapting them according to each user's characteristics and preferences. However, in our initial implementation we only connect the UIM to the dialogue manager, the context manager and the natural response generator. This way, the system can collect information about user's interactions, it can also suggest and execute goals, and it prompts personalized messages to the user.

3 The User Information Manager

To carry out the tasks of identifying users from their speech signal and creating, updating and loading a profile associated to each user, we have developed the UIM. This module is composed of a speaker identification (SID) module and a profile manager (PM) module. We now briefly explain each of these subsystems.

3.1 Speaker identification task

The main goal of the speaker identifier consists of extracting the identity of the speaker of a given utterance. The identification task can be *closed* if the SID has to choose a given identity among a closed set of speakers, or *open*, if the system can assign a new identity tag to an utterance that is not believed to belong to the existing identities. Likewise, the identification task can be done in a *supervised* or an *unsupervised* way, depending on whether a labeled database is available or not. Finally a system can apply a supervised or an unsupervised approach depending on the current and previous dialogue acts.

The SID module we have included into the whole dialogue system takes as input the same sampled speech signal the ASR uses. After a preprocessing stage of extracting the 13 first MFCC of each frame of the speech signal, the SID carries out the identification task by applying a clustering algorithm over the feature vectors.

While (Kinnunen, Kilpeläinen, and Fränti, 2000) makes a comparison between different clustering approaches, our algorithm (Ferreiros, 2000) is different to those. First of all, we have modeled each user utterance as a multidimensional full-covariance Gaussian. The identification task is done by means of grouping different utterances according to a distance measure, following the idea that different utterances of the same speaker will have similar features (namely, MFCC or MFCC and its first order derivatives). That is, the Gaussian distribution for the new utterance will be closer to the Gaussian distribution for the set of utterances of the given speaker than the rest of speakers' utterances.

The chosen metric for the clustering algorithm has been the Bayesian information criterion (BIC), which applies a penalty factor related with the complexity of the used model. As we are modeling each cluster as a multidimensional Gaussian distribution, the complexity of the model is related with the dimension of that distribution ($13 + (13 * 14)/2$ independent elements when using only MFCC, or $26 + (26 * 27)/2$ when including also MFCC first order derivatives).

Our system can carry out both supervised and unsupervised classification approaches. This fact allows us to load labeled databases if we have a set of models of known users. Otherwise, our system can create its own models during each dialogue session, in which different users interact with the SDS.

With each new utterance the algorithm re-estimates the user models associated with each cluster. Since the decision taken by the clustering algorithm can be correct or incorrect, this re-estimation can cause identification errors on future utterances. To avoid that, we have applied a strengthening of our models by means of a housekeeping algorithm, which deletes outlayers from the clusters, and a revision algorithm, which decides if a previously stored utterance could be better classified in a different cluster, due to the effect of other utterances classified after that.

3.2 The Profile Manager

The profile manager (PM) is the subsystem responsible of creating, loading, updating and employing the information associated to the user identified by the SID module. In order to fulfill these different tasks we have defined the contents of the data structure associated to each user according to EDECAN Project[1] proposals.

The information related with a given user is stored in a text file. The PM reads the file associated with the identified speaker and loads it in a data structure called *user profile*. The information of a user profile can be classified into three different groups:

- *General user information.* This set includes personal information, such as user's name, gender, age, current language, degree of expertise when interacting with dialogue systems, pathologies or speech disorders, if any, and so on.
- *General statistics.* This second group comprises the number of sessions, dialogues and dialogue turns, their durations, the date of the last interaction with the system, etc.
- *Usage statistics and user privileges.* This set stores the counts of each action over the system that a user performs, and a mark of user clearance for each possible action.

This last set of statistics is split into different *statistic groups*, which are defined according to the domain-knowledge we have. This new subdivision allows the system to infer the preferences of each user among the different statistic groups, and among the different items belonging to the same group.

As we want to control a Hi-Fi device, an appropriate set of statistic groups is the following:

- *Device*: CD player, tape, radio.
- *CD player*: CD1, CD2, CD3, random playing mode.
- *Radio*: FM, AM, set of different tunes stored on Hi-Fi memory.
- *Cassette*: Tape1, Tape2.
- *Volume*: Very low, low, medium, high.
- *Equalization*: Flat, Soft, Vocal, Heavy, X-Bass.

[1]http://www.edecan.es/

The table 1 shows a translated excerpt of a user profile. We only present information related with the statistics and privileges group, since this is the group of interest when annotating events and suggesting hypotheses. The first column shows the name of the different statistic elements. The second one is the value of each of the previous fields. The third one, if exists, shows the permission of each possible action to be done over the Hi-Fi device.

Statistics and privileges		
CD Player		
CD 1:	3	Allowed
CD 2:	7	Allowed
CD 3:	0	Forbidden
CD random:	0	Forbidden
Equalization		
Xbass:	0	Allowed
Flat:	4	Allowed
Heavy:	0	Forbidden
...	...	...

Table 1: Translated excerpt of a user profile

The initial user profile for a new user can be built manually or automatically. In the first case, a system developer just fills in a file with the information the operator knows about each user. In the second case, the system itself asks the user for his/her static data, included in the set of general information (i.e. name, age, language and so on). Our initial system makes use of the first approach because the baseline dialogue system cannot establish dialogues to get personal information of each user.

When the system identifies a new user it assigns an order number to the identity of that user, and creates a profile with each field of the general statistics and the usage statistics set to zero.

3.2.1 Event annotation and dynamic update.

As a certain dialogue progresses, the UIM has to modify the profile of each identified user in order to update the previous information with that obtained in the last dialogue act. This task can be made via a probabilistic approach (Billsus and Pazzani, 1996), by annotating those features that maximize an expected information gain function. Our initial implementation proposes a simpler strategy. We update the count of actions a user wants to carry out.

Since each count depends on the actions to execute over the final system (i.e. the goals inferred by the dialogue manager that can be achieved), the PM is a strongly application-dependent system. This dependency is reflected on the different statistic groups of actions that can be fulfilled. However, our approach is easily portable to different environments where a SDS could be used.

Our system can deal with explicit and implicit interactions, updating the statistic counts in a proper way. If the user explicitly asks the system for a certain action (e.g. '*play the CD two*') the PM increments the appropriate counts by one.

An example of an implicit annotation is the user acceptance of the initial setup of volume and equalization. When a user starts his/her first dialogue the system updates the counts of these statistic groups in the appropriate user profile.

One of our goals is to adapt the management of user profiles to the evolution of the behaviour of a user throughout his/her dialogue sessions. This can be done by giving more relevance to the most recent dialogue turns. Our first approach consists of applying a proportional reduction over each statistic group at certain moments. Instead of implement time-based alternatives to solve this task we have preferred to apply the reduction to each statistic group when the total counts of elements of each statistic group is equal to or greater than a predefined threshold count.

3.2.2 Hypothesis suggestion.

The event annotation criterion presented in the previous section allows us to speed up the resolution of incomplete dialogues by proposing actions to the user, if his/her associated profile shows a clear preference of that action over the rest of possibilities.

We use the statistic counts to build a model of user preferences, which allows the system to suggest hypotheses in incomplete information situations. We define an *incomplete information situation* as a dialogue flow statement in which the dialogue manager infers that it needs more information than the one inferred during the last utterance of the user in order to fulfill the actions the user has asked for.

To clarify this mechanism we propose the following example. Let's suppose that our Hi-Fi device has a 3 CD unit. If the user says '*play the first track of the CD*', the system needs to know what CD unit the user wants to play. The dialogue manager can obtain that information by taking into account previous utterances stored in the context manager. If knowledge about the CD unit is not available in the CM, the system will ask the user for this information, synthesizing an adequate prompt (e.g. '*what CD unit do you want to play?*').

Instead of directly asking the user, our approach looks first up into the user profile if the user has a clear preference for a certain action. To propose hypotheses, we study the two maximum values of each statistic group. If the quotient between the maximum and the second maximum is higher than a given threshold, the system will assume that the user has a clear preference for the action which has the maximum value of that statistic group. In that case the preferred action will be annotated into the context manager, so that the information about user preference becomes available to the dialogue system.

4 *Assessment of the proposed approach*

4.1 Speaker identification task

We have tested the speaker identification module with two different databases, the CMU ARCTIC database (Kominek and Black, 2004) and our proprietary database, referred to as HIFI-MM1.

The first one consists of 1132 English sentences spoken by 7 speakers (5 male, 2 female), mainly used in speech synthesis research. We have used this database because the recorded material is very close in duration (no more than 4 or 5 seconds for each sentence) and number of speakers (7, almost a maximum number of users at home) to a domotic control application at home.

The HIFI-MM1 database consists of 100 Spanish sentences spoken by 13 speakers (7 male, 6 female). This database is composed of in-domain sentences, that is, sentences that could be used to command actions to the Hi-Fi system.

We have evaluated each database using two different feature sets. In the first one we use only the MFCC, and in the second one we have used both MFCC and their first order derivatives. We also have developed both unsupervised and supervised tests over each database and over each feature set. Our system applies the identification algorithm for each input utterance in order to be close to the behaviour of our final dialogue system. That is, we want to identify the user each time he or she speaks to the system.

The results of these experiments are shown on table 2. As we can see our speaker identification system obtains high identification rates: up to 98% with a supervised approach and a relatively small training set (about 80 utterances per speaker), and up to 73% with an unsupervised approach. Using MFCC and their first derivative yields better identification rates than using only MFCC coefficients in both supervised and unsupervised approaches.

If we compare the unsupervised results with the supervised ones we can estimate the goodness of our clustering approach in terms of how close the unsupervised method results are to the supervised method ones. As table 2 shows, we are especially close to the best achievable results when using a 26-dimension feature space. This fact implies that a relevant quantity of information about the identity of a speaker relies on the first derivative of the MFCC.

4.2 Subjective assessment of the whole system

Concerning the whole system, we have selected different subjective metrics used in (Fernández et al., 2008) to assess the behaviour of our algorithms.

To carry out our evaluation we recorded a database with 3 different speakers, each one saying 82 sentences of our application domain. We use this database to build initial models for the Speaker Identification module. We also generate a user profile for each of those speakers. Instead of building these profiles automatically during several dialogue sessions, we build them manually in order to demonstrate the goodness of the algorithms we have developed. The results showed that the system shows a tendency of answering with higher agility and naturalness than without including the UIM.

As examples of how different the dialogue flow is when including user profiles, tables 3 and 4 show a dialogue without the profile manager unit, and a very close dialogue with our new approach.

We can see how the objective of playing CD is fulfilled with less dialogue turns when including the profile manager, thanks to the information stored in the user profile.

We also see the reaction of the system when a forbidden setting of the Hi-Fi device is asked by a certain user. In our baseline system the users have total control of the system. Now certain functions and objectives can be forbidden, providing the system with a permission control.

5 *Conclusions and future work*

An approach to integrate user-dependent information to modify the behaviour of a spoken dialogue system is presented. This information is used to consider user preferences and characteristics (related with usage permissions), speeding up the dialogue flow in certain dialogues.

Our UIM is transparent enough to the Dialogue Manager. With this constraint, we have modified only the Context Manager to load and update user profiles. Now we can propose several improvements to our initial solution. For instance, we have not applied an adaptation of the acoustic and linguistic models of the speech recognizer, neither have included user preferences concerning different synthesis styles. We have started working on dialogue contextualization, making an important effort in building dialogue-dynamic grammars.

An important issue in which we have started working into is the application of usage permissions. The information associated to each user has a set of tags, each of them indicating whether that user is allowed to use a certain function of the system. A difficulty when including those permissions in a closed system as is a Hi-Fi device is that the system cannot provide us a continuous feedback of the state in which it is. This fact becomes a problem when a new user starts a dialogue after another user has made use of the device. In this situation the initial state of the system for the new user can be a forbidden one.

To deal with the previous problems we are working into two different approaches. The first one consists of keeping several context memories instead the single one which is currently used. This way we expect to associate

	Unsupervised experiments				Supervised experiments			
	ARCTIC		HIFI-MM1		ARCTIC		HIFI-MM1	
	MFCC	MFCC+ ΔMFCC	MFCC	MFCC+ ΔMFCC	MFCC	MFCC+ ΔMFCC	MFCC	MFCC+ ΔMFCC
ER (%)	17.54	0.43	27.86	2.62	3.72	0.43	7.86	2.14

Table 2: Error rates of the speaker identification system

User (John)	Switch the Hi-Fi on.	
System	What do you want to play?	*The system asks the same question to every user.*
U.	Play the CD.	
S.	Which CD unit do you want to play?	*The system does not know the unit to play and this information is not on the context manager, so the system has to ask the user for that.*
U.	CD two.	
S.	Playing track one of CD two.	***Dialogue fulfilled!***
U.	Now play the tape and set the volume to five.	*A new dialogue with two objectives.*
S.	What tape do you want to play?	*The system has to ask the user for the tape to play.*
U.	Play tape two.	*A new dialogue turn is needed to*
S.	Playing tape two. Setting volume to five.	*fulfill user's objectives.*

Table 3: Example of dialogue session without profile manager

User (John)	Switch the Hi-Fi on.	
System	What do you want to play, John?	*The SID correctly identifies John and the PM loads his associated profile.*
U.	Play the CD.	*John's profile shows a preference of playing CD two, so it is loaded into the CM.*
S.	Playing track one of CD two.	*The system suggests and plays CD two.* ***John's acceptance of Hi-Fi settings finishes dialogue.***
U.	Now play the tape and raise the volume to five.	*Two new objectives. John's profile shows no preference about tape ...*
S.	What tape do you want to play, John?	*so the system has to ask John for the number of tape to play.*
U.	Play tape two.	*A new dialogue turn is needed to*
S.	Playing tape two.	*fulfill John's objective.* ***This setting will be annotated on John's profile.***
	I'm sorry, John. You don't have permission to set Hi-Fi volume to five. Please select another volume.	*John's profile shows that 'volume 5' is a forbidden setting for him ...*
U.	Set volume to three.	*so a new dialogue turn is needed.*
S.	Setting volume to three.	***Dialogue fulfilled!***

Table 4: Example of dialogue session with profile manager

to each user both the current state of the Hi-Fi but also the state of the last dialogue of that user. The second one relies on a special treatment of the initial utterances of a user. We try to keep the current state of the Hi-Fi device always updated in the dialogue manager, so that if a new user is identified and the device is in a forbidden state, the DM will generate a sequence of commands that change the state to an allowed one, and will report an appropriate message to the user.

Another task we are working into is the user rejection treatment, specially related to user rejections of system-hypothesized actions. This task will require an increase of the vocabulary supported by the speech recognizer, and the definition of new concepts and goals, modifying thus the Dialogue Manager.

Our initial evaluation shows the potential possibilities of a proper management of user information. However, we have evaluated only a reduced set of subjective features. To obtain more general results we are starting an exhaustive evaluation which will include both objective and subjective metrics. Among the objective ones we will make a special effort on analyzing the number of dialogue turns in which the system makes use of contextual information and the average number of dialogue turns since we expect that the correct identification of the speaker together with the proper hypothesis suggestion will improve those metrics.

References

Billsus, D. and M. Pazzani. 1996. Revising User Profiles: the Search for Interesting Web Sites. In AAAI Press, editor, *Proceedings of the Third International Workshop on Multistrategy Learning (MSL'96)*.

Eckert, W., E. Levin, and R. Pieraccini. 1997. User Modeling for Spoken Dialogue System Evaluation. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU97)*, pages 80–87.

Fernández, F., J. Blázquez, J. Ferreiros, R. Barra, J. Macías-Guarasa, and J.M. Lucas-Cuesta. 2008. Evaluation of a Spoken Dialogue System for Controlling a HiFi Audio System. In *Proceedings of the IEEE Workshop on Spoken Language Technology (SLT08)*, pages 137–140.

Fernández, F., J. Ferreiros, V. Sama, J.M. Montero, R. San-Segundo, and J. Macías-Guarasa. 2005. Speech Interface for Controlling a Hi-Fi Audio System Based on a Bayesian Belief Networks Approach for Dialog Modeling. In *Proceedings of the 9th European Conference on Speech Communication and Technology (INTERSPEECH05)*, pages 3421–3424.

Ferreiros, J. 2000. Acoustic Change Detection and Clustering on Broadcast News. Technical report, International Computer Science Institute, Berkeley, USA.

Kinnunen, T., T. Kilpeläinen, and P. Fränti. 2000. Comparison of Clustering Algorithms in Speaker Identification. In *Proceedings of the International Conference on Signal Processing and Communication (SPC00)*, pages 222–227, Marbella (Spain).

Kominek, J. and A.W. Black. 2004. The CMU ARCTIC Speech Databases. In *Proceedings of the 5th ISCA Speech Synthesis Workshop*, pages 223–224, Pittsburgh, PA.

Pargellis, A.N., H.K.J. Kuo, and C.H. Lee. 2004. An Automatic Dialogue Generation Platform for Personalized Dialogue Applications. *Speech Communication*, 42:329–351.

Schatzmann, J., M.N. Stuttle, K. Weilhammer, and S. Young. 2005. Effects of the User Model on Simulation-Based Learning of Dialogue Strategies. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU05)*, pages 220–225.

Schatzmann, J., M.N. Stuttle, K. Weilhammer, and S. Young. 2006. A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies. *The Knowledge Engineering Review*, 21:97–126.

Webb, G.L., M.J. Pazzani, and D. Billsus. 2001. Machine Learning for User Modeling. *User Modeling and User-Adapted Interaction*, 11:19–29.

Zukerman, I. and D. Litman. 2001. Natural Language Processing and user Modeling: Synergies and Limitations. *User Modeling and User-Adapted Interaction*, 11:129–158.