Universität Karlsruhe (TH)

Diplomarbeit

Social User Model Acquisition through Network Analysis and Interactive Learning

Diplomand: Felix Putze
Betreuer: Prof. Dr. A. Waibel
Betreuer: Dipl.-Inform. H. Holzapfel

Tag der Abgabe: 29.02.2008

Wintersemester 2007/2008

Fakultät für Informatik
Institut für Theoretische Informatik (ITI)

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbständig verfasst habe und nur die in der Arbeit angegebenen Hilfsmittel und Literaturstellen verwendet habe.

Karlsruhe, den 29. Februar 2008 F. Putze Felix Putze

Contents

1. Introduction & Motivation

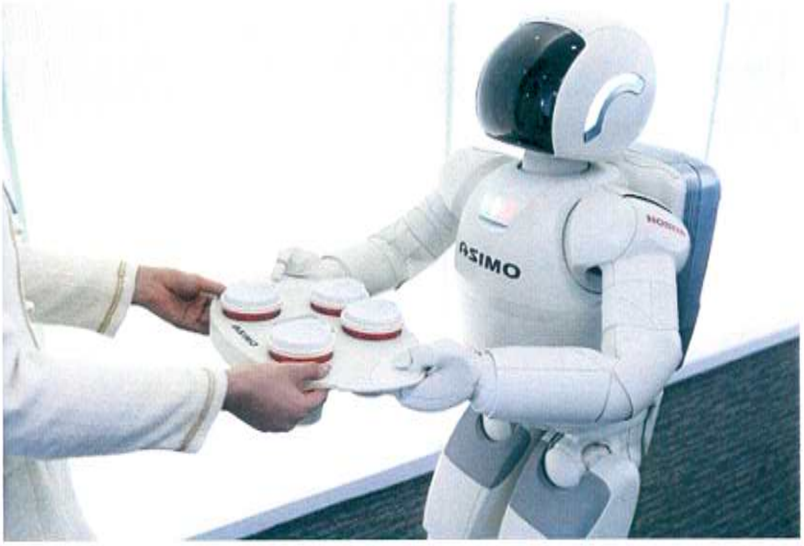
1.1. Humanoid Robots & User Models

Interactive humanoid robots are studied with increasing interest and effort [4, 28]. A typical example of this class of robots is ARMAR [4], a humanoid robot designed to assist humans in a kitchen environment.

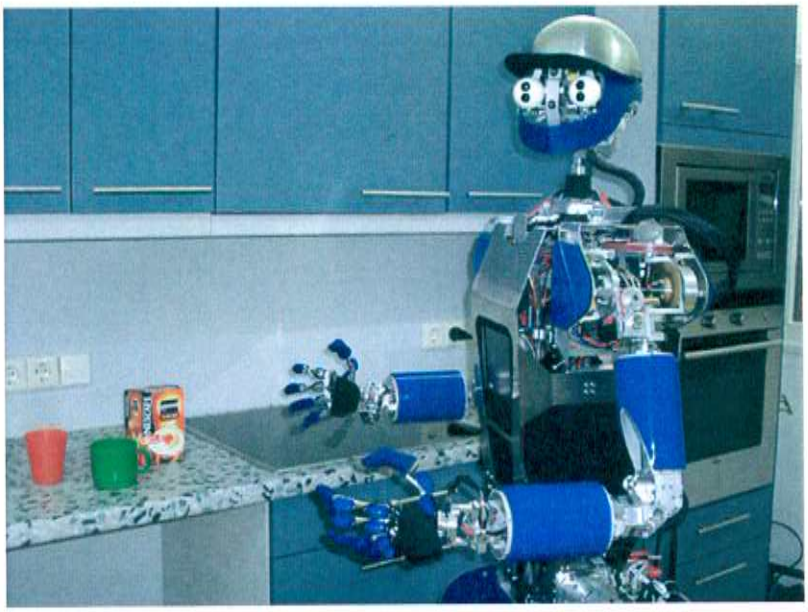
A humanoid service robot is meant to work among people and to integrate in their daily life. For the users to fully accept the robot as a companion, it has to be aware of the emotional states and social relations among people. This knowledge would allow the robot to behave accordingly during interaction with humans, for example to back up if the user is afraid of the massive robot or to deny the children unauthorized access to the candy box. It also helps the robot to better understand human-human interactions. In general, learning more about its users will make any interactions with humans more natural. This work concentrates on the social aspect of user modeling.

State-of-the-art systems model a user's name, his face and his voice [19, 24]. Theories in psychology indicate, however, that human identity consists of more aspects. A person's social ties, his interests and his role in society are part of his identity, influence his behavior and the behavior of others towards the person during social interactions. In this work, we are building *social* user models. These user models enable the robot to model, update and request social information on its current user or other humans it is referring to. This contains, but is not limited to, the users' roles, their relationship among each other and their interests.

The scenario in which social user models are learned (called *IslEnquirer*) consists of a multimodal humanoid robot in a corridor of the Interactive Systems Lab (ISL). This setup was used before for human-robot receptionist dialogs [20]. It is now extended by us to a system working autonomously for several consecutive sessions and which does not only learn the user's name but also builds a social user model by interviewing the user about his relations and interests concerning his work. We concentrate



(a) Asimo



(b) ARMAR 3

Figure 1.1.: Examples of advanced humanoid robots

on ISL staff members and students, but other persons are learned as well to deal with regular guests.

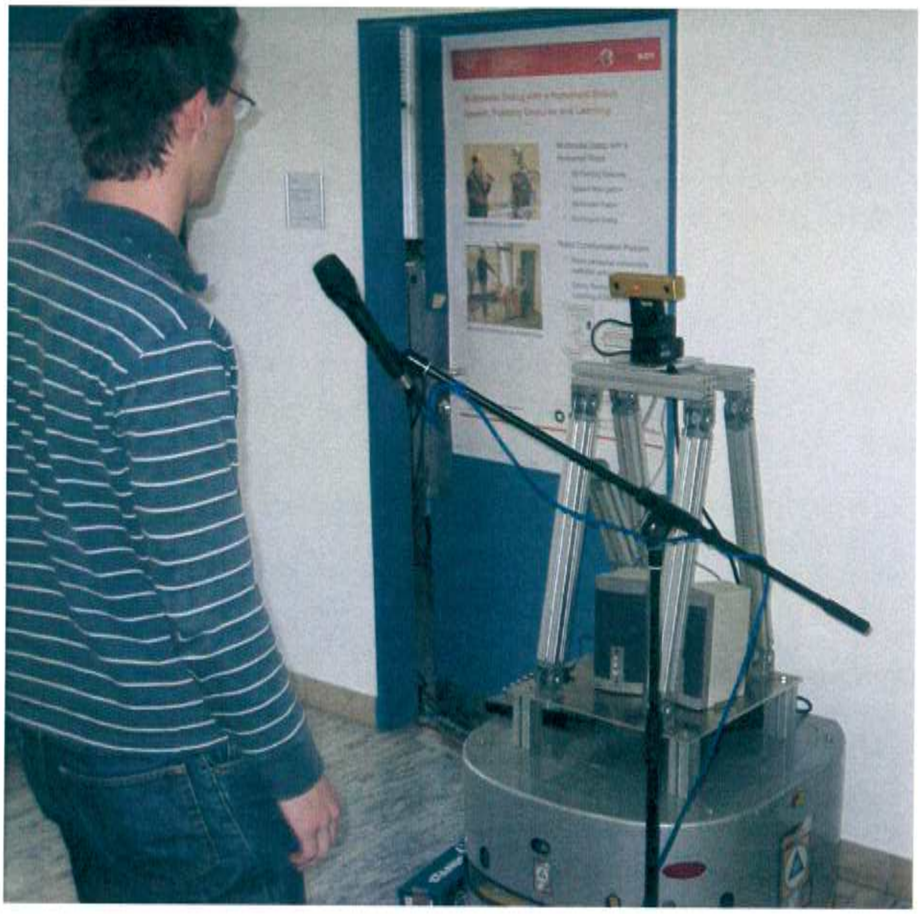


Figure 1.2.: Robbi in action

To support the robot in its task with a reasonable foundation of knowledge, we begin with an *offline step*, i.e. an analysis of available data. For our academic scenario this is a corpus of publications which we can easily access from the web site of the institute. We automatically analyze this data with methods of social network theory to create initial hypotheses on the degree of cooperation among the members, the groups they are organized in and their roles.

To present the results, a dynamic web site is generated from the gathered data. This site will always be up-to-date in contrast to one which is maintained manually (cf. figure 1.3), especially if there is no designated administrator. In addition, the data will be acquired through automatic data analysis and short human-robot dialogs, which means that no typing is required to update the web page. As the system uses different sources, we can expect it to be more objective and unbiased.

Publications

Here are all the publications:

2006

HCRS

- Christian Fügen, Hartwig Holzapfel, Florian Kraft
  Natural Human Robot Communication
  Human Centered Robotic Systems, HCRS, München, Germany, 2006-10
  (pdf)

Interspeech 2006

- Sebastian Stüker, Christian Fügen, Susanne Burger, Matthias Wölfel
  Cross-System Adaptation and Combination for Continuous Speech Recognition: The Influence of Phoneme Set and Acoustic Front-End
  International Conference on Spoken Language Processing, Interspeech 2006, Pittsburgh, PA, USA, 2006-09-00
  (pdf)

- Christian Fügen, Matthias Wölfel, Shajith Ikbal Mohamed, Florian Kraft, Kornel Laskowsk Mari Ostendorf, Sebastian Stüker, Kenichi Kumatani
  Advances in Lecture Recognition: The ISL RT-06S Evaluation System
  International Conference on Spoken Language Processing, Interspeech 2006, Pittsburgh, PA, USA, 2006-09-00
  (pdf)

Figure 1.3.: A screenshot of the ISL website from summer 2007. Note that many publications from this year are missing and that some researchers are overrepresented.

1.2. A Vision of a Socially Aware Robot

Why is it helpful to have a robot learn social relationships and interests? The following section presents a vision of a robot that profits from such knowledge by providing a more natural and individual dialog experience to the user. These ideas are beyond the scope of this work and probably beyond the state-of-the-art in humanoid robots in general. However, some of them have already influenced the design of the IslEnquirer. We believe that they provide an insight in how useful it is to acquire social user models:

We think of a robot that interacts with humans through speech. It is able to adopt its dialog behavior in terms of style, structure and content to the current user. For example, knowing the interests of a user can help to determine which information he is interested in. In our academic environment, the system could propose newly published papers of the user's field of research. If the system needs to extend or update its database on a certain topic, it can decide whether the user is likely to possess such knowledge.

A robot that is aware of the social network between its users can estimate whom certain news have already reached and who still needs to be informed. This knowledge also enables the system to ask the current user questions on third persons who the robot does not see regularly.

Not only the content of a dialog depends on the participants but also its structure. A busy professor will be interested in a precise and concise dialog, while a first-time guest will expect a warm welcome and some chatter while waiting for his host.

For another application, consider a dialog system used for participant registration at a large conference. There will possibly be people sharing the same name or the speech recognition component cannot decide between two similar names. Further knowledge about the participants could help the dialog system to easily distinguish between different persons by asking for alternative attributes like role or research area: "Are you a professor researching on speech recognition?". This could often avoid asking to repeat or spell the name.

1.3. Goals & Challenges

The first goal of the IslEnquirer has to be to investigate if it is possible at all to gather social information using human-robot interaction. As we will see, even humans struggle or disagree on explicitly modeling social information. A robot with its limited speech recognition and understanding capabilities might have even more problems. Our experiments show that we were able to find satisfying solutions these problems.

Once we created the possibility to gather social information, we begin with an offline analysis to prepare the robot for interaction with humans. The goal of this step is to acquire hypotheses for as many social attributes as possible. To do so, we use established and new methods from social network theory and information retrieval. At the same time, we need to extract a vocabulary for speech input and output. This vocabulary enables the system to actually refer to and understand references to abstract social attributes like groups or roles. We restrict ourselves to a fully automatic system in this step as we aim for a chain of components that can collect information completely on its own.

The next goal is to design a dialog system for the robot which is capable of extracting new information and verifying existing data. The challenge here is to gather as much and as reliable information as possible with as few questions as possible. The robot has to keep the dialog interesting and varied for the user. This means a regular change of topic and the use of different types of questions. The more often a user is seen, the less he should be bothered with questions. The solution to this problems is to define a question selection algorithm that takes all these requirements into account and balances them.

The system will be confronted with unreliable data and noise as a result of speech recognition errors, misunderstandings and wrongly informed users. Furthermore, the system must process information from different sources and information of unknown quality. All three points suggest that we need a representation for our user model that is robust against noise, can deal with multiple hypotheses and give a confidence measure to compare those hypotheses.

Another goal of the IslEnquirer system is to form a stable basis for future human-robot interaction scenarios. This makes it necessary to build a system that can run for several hours without intervention of an operator. It must be possible to quickly replace parts of the system to adapt it to a new task.

2. Related Work

2.1. User Models

The theory of social identity [35] assumes that a person has not a single "personal self". Instead, its personality and behavior depend on the social context in which it is acting. The authors of this work claim that in certain situations, a person identifies with a social group it belongs to. This helps to bolster the person's self esteem and to sharpen its self image by comparing to and disassociating from other groups. The definition of social groups is not fixed in size or composition. The importance of this theory for our work lies in the observation that social affiliation is a major influence for human behavior and should not be ignored by humanoid robots.

[8] and [15] coin the terms of group-structural awareness and social awareness in the context of human-robot interaction. According to [8], group-structural awareness acquires knowledge in terms of a person's role and responsibilities, its position on an issue, its status, and group processes. Social awareness is awareness for the understanding that participants have about the social connections within their group and information about the presence and activities of people in a shared environment. This is the kind of awareness we aim for by building social user models.

User models in general are an important component of successful human-computer interaction. [11] claims that the challenge of designing interactive systems in an information-rich world is not only to make information available to people at any time, at any place, and in any form, but specifically to say the right thing at the right time in the right way. This means that such a system, especially a humanoid robot, must adapt to the users background and behavior to provide maximal usability.

[37] defines a user model as a knowledge source in a natural-language dialog system which contains explicit assumptions on all aspects of the user that may be relevant for the dialog behavior of the system. [23] refines this by formulating four requirements to user models: *separate knowledge*

base, *explicit representation*, *support for abstraction* and *multiple use*. The authors also give four categories of information that is stored in user models: *goals and plans*, *capabilities*, *attitudes*, and *knowledge* or *belief*. The contents of a social user model fall into the latter category as it models the users' beliefs in their social identity. By using the terms "knowledge" and "belief" for the same category, the authors claim that both are interchangeable from the perspective of the system as it cannot distinguish whether a proposition is known or simply believed to be true. This is important to notice as it motivates a user model that incorporates information from multiple sources and sources that are independent of the single user.

[41] presents one of the first approaches to active user model acquisition by querying the user during a spoken dialog session. To decide whether and how the user model is actively updated, the author proposes to use decision theory to weigh several criteria against each other, e.g. the likelihood of success or the expected productiveness of a acquisition subdialog. Similar decisions will also be guide the development of the IslEnquirer dialog component.

To our knowledge, there exists no social user model for human-robot interaction. The term *role* for example usually refers to short-lived roles taken by the involved actors during one dialog session: Typical roles for the human are operator, mechanic, teammate or bystander [33].

[19] describes a humanoid robot capable of learning new and recognizing known people to allow personalized services. The system acquires knowledge on the face, the voice and the name of the user, which form the user model of the system. All information can be gathered automatically without precollected data and the robot is designed to automatically attract people to initiate a dialog. This work successfully derives user information solely from the interaction. It forms the basis for the interactive part of the IslEnquirer system and will be extended to also learn social user models.

2.2. Social Networks & Information Retrieval

Social network analysis and information retrieval are both very active fields of study and reviewing all relevant literature is beyond the scope of this work. We restrict this overview to components that either influenced the design of the IslEnquirer system or which present similar applications.

The classical works on social network analysis in the context of academic publications are [12, 40]. [12] was one of the first works on studying the networks induced by co-authorship and citations in scientific publications. It uses automated methods to analyze the structure of these networks and to identify the most important publications and journals. [40] is the work that introduced the formalization of roles in a graph theoretic way.

More recent work on the algorithmic analysis of social networks is found in [2, 3, 26]. The authors of [2] use co-authorship of publications to establish a social network of the scientific community and study the evolution and topology of this complex system. The authors use empirical and analytical measurements and numerical simulations to analyze scalability, connection degree and other characteristics.

[3] proposes to analyze roles and social positions across different large internet-based communities to identify similar user behavior. The paper surveys and analyzes formalizations of positions within the network with a special emphasis on the use of algebraic notions.

[26] presents an approach of name disambiguation using social networks. The author shows how in a large movie actor database several people that share the same name can be seperated. This is done by building a social network in which each occurrence of an ambiguous name is represented as a single node. The author then uses random walks to determine a similarity of neighbors between two nodes.

[42] describes the Arnetminer project, a web site presenting automatically gathered information on members of the worldwide scientific community. This information is, among other, the person's affiliation, the research interest and a list of associated researchers. The system searches the web and popular publication indices for data and employs a combination of several classifiers and heuristics to extract the relevant information from the found web sites and publications. Based on co-authorship, the authors built a social network to identify cooperations within the community. The scope of this work is much wider than the one of the IslEnquirer scenario and it does not report research groups or social roles.

The authors of [31] build a document recommendation system which integrates a content-based and a social-based component. For both parts, a user model is constructed. The content-based component compares documents in its database to the user's interests via TF-IDF scoring (cf. section 3.8.6). The interest is estimated by modeling the content of the previous documents requested by the user. The social component uses explicit or implicit votes of other users to rate the documents. The votes are weighted by a similarity score which is calculated for each pair of users. The authors test their system for movie recommendation but claim its domain independence. Transfered to the IslEnquirer nomenclature, this work builds a social user model containing the attribute *interests* and an *importance* measure of one person for another. Like the IslEnquirer, it combines methods from information retrieval and user interaction. However, the interaction component works by voting and is much less explicit than a spoken dialog based approach.

3. Basics & Terminology

3.1. Spoken Dialog Systems

A modern dialog system consists of multiple building blocks which form a sequence of components that allow the system to parse, understand and react to the user's input. This chain will be run through every turn, i.e. for each user input and corresponding system output. An overview is shown in figure 3.1 and all components will be described hereafter. The most important parts will then be explained in greater detail in the following sections.

- **ASR** Any user input is recorded using a close-talk microphone. A segmenter determines the boundaries of a spoken utterance based on signal characteristics like energy level and signal-to-noise ratio. The recorded audio is then sent through an automatic speech recognition system (ASR). This component determines the most likely textual representation for each spoken utterance.
- **NLU** The natural language understanding unit (NLU) converts the input string to a semantic representation called Typed Feature Structure (TFS) using a semantic grammar. This representation is independent of the specific phrasing. If the system can process multiple modalities (e.g. gestures, emotion), these would have their own understanding unit and result in compatible semantic representations which are independent of modality.
- **Context** The TFS is then interpreted within the context of the current utterance. For example, if the user input is of the type `confirmation`, the system has to convert this response to a name confirmation, the approval of continuing the dialog or any other semantic concept, depending on the context of the system utterance the user is referring to.
- **Discourse** Any non-trivial dialog consists of multiple turns. This means that the newly extracted TFS has to be brought in accordance

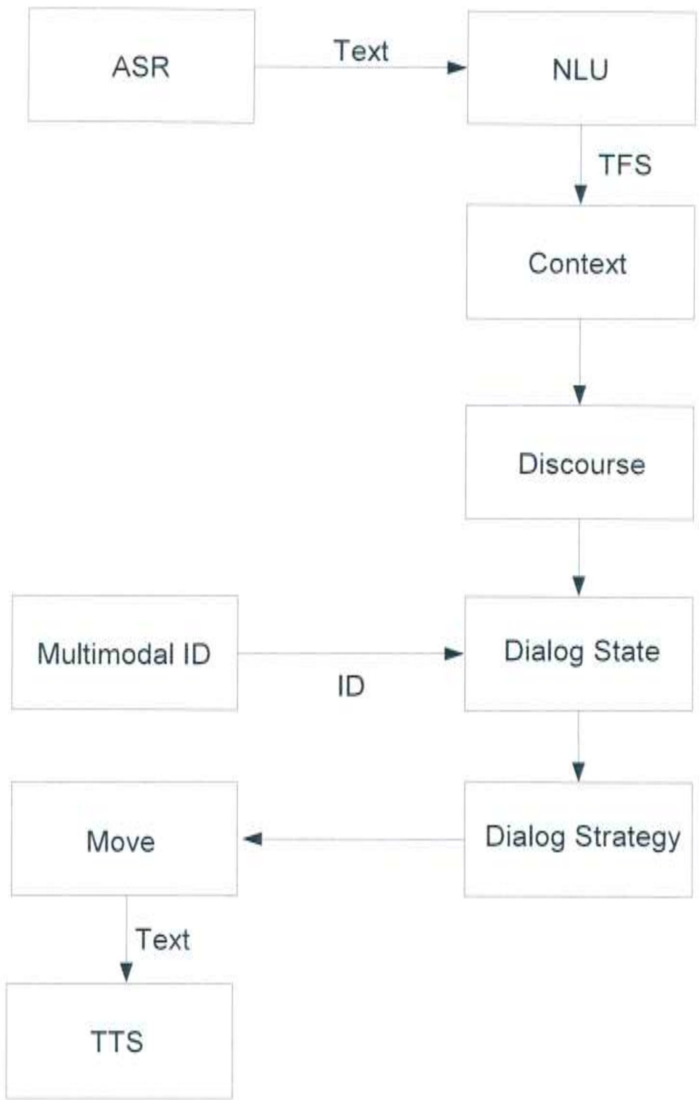


Figure 3.1.: The main components of a dialog system

with the entirety of all previously acquired semantic concepts. This combined history of dialog is called discourse. It usually reflects the dialog structure up to this point and any open dialog goals.

- **Dialog State** The discourse together with an array of internal variables (e.g. number of passed turns) forms an abstract dialog state. This state is the basis for deciding on the next system action. It can also include the result of other components like the hypotheses list of the user identification unit.
- **Strategy** The dialog strategy takes the dialog state and an explicit or implicit task model to determine the next system action (*move*). Which criteria this decision is based on and which decision rule is applied is implementation specific.
- **Move** A move is an abstract system action that can consist of more than one atomic action. Examples for atomic actions are outputting speech, writing to a database or learning a new word.
- **TTS** Almost every move generates a system response or a new prompt represented by a text string. The job of the text to speech (TTS) component is to convert this string to audible speech.

To ensure reliable communication between all components, a messaging middleware is used. The IslEnquirer components make use of the One4All (O4A) communication protocol. This lightweight communication server allows transfer of short arbitrary messages. Another benefit of O4A is synchronization between components, for example to coordinate the start of a new session.

3.2. Automatic Speech Recognition

A modern ASR system is a statistical framework which estimates the most probable candidate $\hat{W}$ of all possible word sequences W for a given sequence of feature vectors X which are extracted from the user's utterance:

$$\hat{W} = \arg\max_{W} P(W|X) = \arg\max_{W} P(X|W)P(W) \tag{3.1}$$

The class based probability $P(X|W)$ is modeled by the acoustic model, which in modern systems uses Hidden Markov Models (HMMs) of single

phonemes. Those HMMs are concatenated to form a search space which is traversed using the Viterbi algorithm.
The a-priori probability $P(W)$ is modeled by the language model. Most spoken language dialog systems use a context free grammar and assign probability of one to each phrase which is covered by the grammar and zero otherwise. Grammars allow to cover the limited domain that most systems are employed in and require very little training data as they can be constructed by a domain expert examining recorded dialogs. As language understanding and extracting semantics from user utterances requires parsing, which is usually also done using a grammar, this type of language model has the additional advantage to only return parsable recognition hypotheses. The coupling between dialog management and grammar based speech recognition can even be extended further [10] if we use the expectation on the user's reaction to a system utterance to weigh grammar rules.
Most general purpose recognizers nowadays use a n-gram language model which statistically estimates the probability of word sequences of length n. The probability of a sentence is the product of the probabilities of all contained n-grams. For more details on ASR, refer to [21].

The Janus recognizer [34] used for the IslEnquirer system supports adaptation of the acoustic model to the current user and detection of unknown words, also called out-of-vocabulary words (OOV). It also features a two-step recognition for certain word classes which allows to switch from a small primary vocabulary to a larger secondary one when an OOV term has been recognized.

3.2.1. Spelling Recognizer

The IslEnquirer system makes use of a spelling recognizer as a device to receive an exact orthographic representation of unknown names and terms. It is typically employed if OOV words are detected and asking for repetition did not improve the result. While spelling is not the most natural modality, it is the most reliable way of teaching the system unknown names and words [44]. A spelling recognizer is a regular HMM based speech recognizer with specialized models: instead of composing longer words by using short phoneme units, we have one HMM model for every letter. The language model is a long range n-gram model. For our application, it was trained on first and last names extracted from German phone books.

3.3. Natural Language Understanding

In this section, we will concentrate on the NLU component of the Tapas dialog toolkit. Many presented concepts can be carried over to other NLU units.

All inputs to the dialog system are transformed into a semantic representation called *typed feature structure* (TFS) [5]. A TFS is a collection of feature-value pairs, describing attributes of a semantic concept. Figure 3.3 gives an example of a prototypical TFS representing the speech act of telling a person's name. It shows that attributes, which are given names in capital letters, can either be atomic values or other TFS. Each TFS is of a certain type, which is defined in an ontology. This ontology is designed hierarchically and supports inheritance. Compatible typed feature structures can be merged using unification. This results in a new TFS in which every attribute value is the more specific value from both input structures. Unification allows to bring together semantic inputs from different sources, which can result in mutual disambiguation.

Figure 3.2 shows an excerpt of a JSGF grammar (in this form introduced by [7]) for a name learning dialog. It consists of context free rules separated by semicolons. Non-terminals are marked with arrow brackets and those marked as `public` function as start symbols. The identifier of a non-terminal consists of the element from the ontology this rule refers to, a grammatical part of speech tag and an additional name. [39] offers more information on the used grammar format. As both Janus and Tapas use the same grammar, the ASR can directly send its parse tree to the dialog manager which then can easily extract the required semantic concepts. This avoids parsing the utterance twice.

The goal of language understanding is to extract semantic concepts from the user's utterance. In the Tapas dialog toolkit, this is done using a semantic grammar which is basically a context free grammar amplified by semantic tags. Those indicate which part of the parse tree is mapped to which concept. Figures 3.3 and 3.2 show an example of how semantic grammars and TFS representation interact: the terms in braces indicate to which attribute of a TFS the tagged grammar nodes belong.

The grammar can read the content of non-terminal nodes from a database to flexibly extend and modify the set of parsable utterances without manual editing of the grammar. This is important to our scenario in which we want to automatically generate the list of descriptive labels a user can use. It is also possible to extend the grammar at runtime: a new terminal is added to the Tapas grammar and for the speech recognition part, the new term is inserted into the language model class that corresponds to the extended non-terminal.

```
public <informName,VP,_> =
↪ <inform_name,V,_> <obj_person,NP,_> {PERSON obj_person};
<inform_name,V,_> =
↪ my [first] name is | i am | you can call me;
<obj_person,NP,_> =
↪ <dbimport_first_name> {FIRST_NAME VAR:NODE_VALUE};
```

Figure 3.2.: Example of Tapas grammar rules.

```
class obj_person inherits generic:object {
↪ base:string :  FIRST_NAME;
↪ base:string :  FAMILY_NAME;
};

class inform_name inherits generic:inform {
↪ obj_person :  PERSON;
↪ base:boolean :  CONFIRM_NAME;
↪ base:boolean :  CONFIRM_FAMILY_NAME;
};
```

Figure 3.3.: Simplified excerpt of the IslEnquirer ontology

3.4. Text to Speech

There exist two main types of speech synthesis systems: Concatenative ones combine short samples of natural speech, smooth the transitions and then add prosody. This usually results in a human-like voice but requires large amounts of audio data. The other group of speech synthesis systems generates a voice by adjusting model parameters trained on recorded speech. This produces voices from a comparably small amount of data, however the voices often sound robotic and unnatural.

The IslEnquirer uses the Cepstral[1] TTS system. The required translation from graphemes to a phonetic representation often is done auto-

[1] http://www.cepstral.com

matically or can be looked up in a hand-crafted dictionary. The latter is often necessary for proper names as their pronounciation deviates from the usual rules of the target language.

The TTS output can be adjusted at runtime by using a standardized, XML-based markup language called *Speech Synthesis Markup Language* (SSML)[2]. This allows to tune the speech generation without accessing the voice or the vocabulary itself. An example of importance for the IslEnquirer system is to tag certain text segments with `emphasis`. The observed effect depends on the used TTS system and voice but usually it will result in slower, more pronounced speech. This makes names easier to understand.

3.5. Dialog Modeling

A dialog system uses a variety of models to describe the discourse, the state of dialog and all other parameters the dialog strategy depends on. The number and type of models are depends on the system, but in the following we give an overview over the most common ones, which are also included in the IslEnquirer system:

- **State Model** The state model is a collection of variables describing the state of the dialog so far. Usually, we have counters for events, for example how often the user's name was already spelled, and boolean flags that store whether a certain point in the dialog flow has already been reached, for example if the user's first name has already been confirmed.

- **Slot Model** The slot model contains named slots that are designated each for storing one atomic piece of information required during one session. The number and content of these slots of course depend on the application. The entirety of all slots describes the gathered knowledge of the system which it can base further dialog moves on and which it can use on (sub)task completion to pass on to another application.

- **Session Model** In the simplest case, the session model is just a boolean variable storing whether a session is running or not. When the system is intended to run autonomously for several sessions, it

[2] `http://www.w3.org/TR/speech-synthesis/`

is likely to include information on when the next session may begin or control the training process when a session is finished.

- **User Model** The user model stores information on the current user, acquired from the user input or by other means of perception. In many systems, this includes the user's identity. The IslEnquirer additionally builds *social* user models which also describe the user's relation to other people, his interests and other attributes.

3.6. Dialog Strategy Types

3.6.1. Finite State Machines

The most simple way of formalizing a dialog strategy is using a finite state machine. A finite state machine models dialog states connected by transitions that are triggered by a certain user action. Each transition in turn can trigger a system action. Finite state machines have the advantage that any domain expert, even if not familiar with the details of dialog modeling, can easily create strategies by using graphical tools. Building finite state machines does not require any data collection and the designer has full control of the complete dialog flow at all times. However, finite state machines also have major drawbacks: In every state, only the user actions intended by the designer will advance the dialog. This often restricts the user in his decisions. A dialog based on finite state machines will therefore likely be very structured and unnatural, leaving little initiative to the user.

3.6.2. Dialog Grammars

As finite state machines accept formal languages of Chomsky-3 type, it is natural to move up to more expressive models. Chomsky-2 languages are generated by context free grammars. A typical rule of a dialog grammar would be
`<questionTurn>` → `<question> <answer> [<confirmationTurn>]`.
Inner non-terminal symbols can be used to model more abstract concepts like initiative. Dialogs based on context free grammars are still very structured.

3.6.3. Information Based Systems

Information based dialog strategies try to avoid the problems of finite state machines by using a more implicit state representation: The discourse is represented by one or more *goals* containing one or more slots. Each slot stores one atomic piece of information, for example a name. The entirety of all slots forms the dialog state, in some more advanced approaches together with the state of the goals which can be selected, deselected or finalized. The dialog designer now creates several *moves* that check for conditions of this dialog state and trigger system actions. Information based strategies make it harder to design the exact dialog flow, but allow to factor out actions that only depend on part of the dialog state. In contrast to finite state machines, every slot can still be filled at any time by the user. One utterance can even fill multiple slots in one step, for example if the user gives both his first and last name at once.

3.6.4. Statistical Systems

Statistical systems employ some kind of statistical learning of a strategy. One approach is to regard dialog management as an optimization task of an agent moving through the space spanned by all possible dialog states while optimizing some reward. This is formalized by modeling the dialog as a Markov decision process or a partially observable Markov decision process. This model is then used to employ reinforcement learning techniques to automatically train a strategy that maximizes the expected long-term reward [30, 43]. Learning of optimal strategies is very appealing on first sight, but the designer still has to put much effort in the proper definition of a state space, available system actions and a reward function. As the optimality of reinforcement learning requires many iterations, real recorded data will usually not suffice to cover the whole state space. Instead, user simulations are trained on the recorded data to produce an infinite number of artificial dialog sessions to train the strategy on.

3.7. Multimodal Person Identification

The IslEnquirer dialog system depends on acquiring the user's identity to assign the gathered information to the right models and to determine for which topics the user can probably provide reliable information. Furthermore, it is necessary to use the user's name in a consistent way without (much) orthographic variation, to keep the database non-ambiguous. Both goals are supported by using a multimodal user identification component that does not only rely on the user saying or spelling his name but also uses visual and acoustic features to identify him. The IslEnquirer system employs a new identification module that integrates a speaker ID using gaussian mixture models on phonetic cues [22] and face recognition using discrete cosine transform [9]. The latter is integrated within the Arthur framework[3] that also provides person tracking and face detection. The identification component regularly queries both modalities and calculates a sequence hypothesis based on multiple consecutive face and voice hypotheses. Both results are then integrated in multimodal (sequence) list of hypotheses together with a confidence value [14].

3.8. Social Network Analysis

3.8.1. Social Networks

Social network analysis is a discipline of sociology that focuses on relations between persons or other social entities. A social network describes *actors* which are connected by relational ties. Mathematically, a social network can be described as a weighed and directed graph $G = (V, E)$ where the set of nodes V ($|V| = n$) corresponds to the actors constituting the network and the edges E together with the edge weight function w describe the ties among the actors. For some applications, it is useful to represent the network with its adjacency matrix A, where $a_{ij} = w(i, j)$. Both formalizations together allow us to apply all knowledge from graph theory and linear algebra for the analysis of social networks.

3.8.2. Cliques & Variants

A clique is a maximal subgraph in which every pair of nodes is connected with an edge of weight > 0. If we want to exclude weak ties between nodes, we can increase this threshold. Finding cliques is known to be NP-complete but as our social network is small and we are processing it offline, this can be handled by using a standard graph library like JGraph[4].

We want to employ cliques to identify groups of closely related researchers. As [38] notes, cliques are a very strict concept of modeling

[3] `http://isl.ira.uka.de/~nickel/arthur/`

[4] `http://jgrapht.sourceforge.net`

cohesive subgroups as no missing links are allowed. There exist some generalizations and relaxations [1] that try to give a less restrictive concept of groups:

A n-clique is a maximal subgraph in which every pair of nodes is connected by a path of length $\leq n$. An equivalent definition which also works for weighed graphs is a maximal subgraph with a diameter $\leq n$. A 1-clique is a regular clique. Usually $n = 2$ is chosen as a suitable cut-off.

A k-plex G_{k-plex} is a maximal subgraph in which for every node, there exists an edge to at least $|G_{k-plex}| - k$ other nodes in G_{k-plex}. Choosing $k = 1$ again results in the definition of a regular clique. This shows that a k-plex is indeed a generalization of a clique.

Figure 3.4 shows all three types of subgraphs in an exemplary graph. The solid lines connect the nodes which belong to the maximal clique of the graph. If we add the dashed lines, we end up with the maximal 2-plex and adding the dotted lines as well results in the maximal 2-clique of the graph. The latter already spans the whole graph in this example.

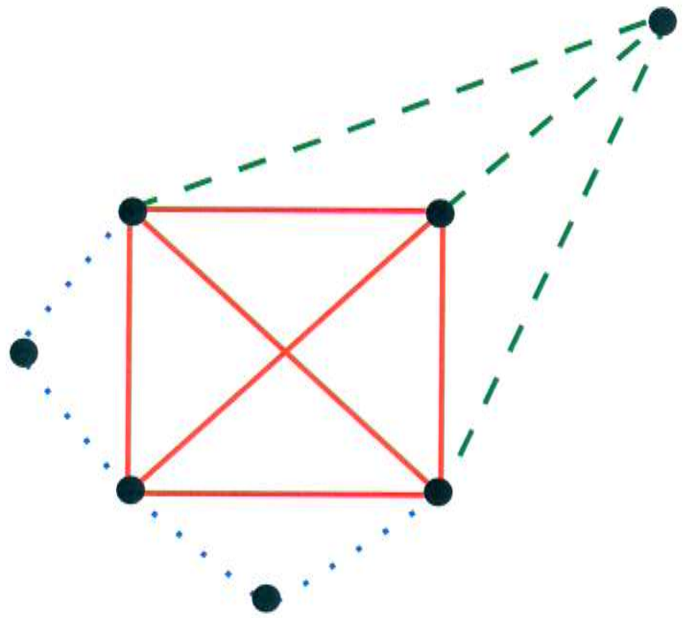


Figure 3.4.: A graph and its maximal clique, 2-clique and 2-plex

3.8.3. Roles

The notion of social roles is a centerpiece of sociological theory. It is at the same time a very difficult concept [16]. The basic idea is that two actors have the same "position" or "role" to the extent that their pattern of relationships with other actors is the same. Nodes in a social network representing people with a common role are called *equivalent*. However, social scientists still have problems agreeing on what relationships need to be examined or how similarity between patterns should be measured. There are several established definitions [38]:

- **Structural equivalence** of two actors is given if and only if they have identical ties to and from the same actors. Structural equivalence is the oldest definition of equivalence and still widely in use. However, it does not generalize well as two equivalent nodes need to share the same set of neighbors. This makes two professors from different universities non-equivalent, as they relate to different research assistants and students.

- **Automorphic equivalence** was proposed as an relaxation of structural equivalence. Here, we call two nodes v and w equivalent if and only if there exists a graph automorphism that maps v to w. Note that structural equivalent nodes are also automorphic equivalent by the mapping that swaps both nodes and leaves the rest unchanged. This definition allows the comparison of nodes across different social networks, but still constraints equivalent nodes to have the same degree.

- **Regular equivalence** is defined recursively: Two actors are regularly equivalent if they have identical ties to and from equivalent actors. For example, two professors are equivalent if we assume that their associated research assistants are equivalent.

The three definitions go from most restricted to most flexible and at the same time from easiest to hardest in terms of identifying equivalent nodes. Figure 3.5 shows an exemplary social network. Intuitively, we can identify three levels of hierarchy. However, this graph contains only two pairs of structurally equivalent nodes: $\{5, 6\}$ and $\{8, 9\}$. The existance of seven different equivalence classes shows a lack of generalization. Automorphic equivalence results in the following partition: $\{1\}$, $\{2, 4\}$, $\{3\}$, $\{5, 6, 8, 9\}$, $\{7\}$. The number of classes is reduced, but still not all nodes of one hierarchy level are equivalent due to different degrees. Finally, the natural partition $\{1\}$, $\{2, 3, 4\}$, $\{5, 6, 7, 8, 9\}$ defines classes of regularly equivalent nodes. However, this partition is not unique: *All* partitions given for this example group regularly equivalent nodes together. In practise, one is usually interested in the *maximum regular* partition, i.e. the most coarse one.

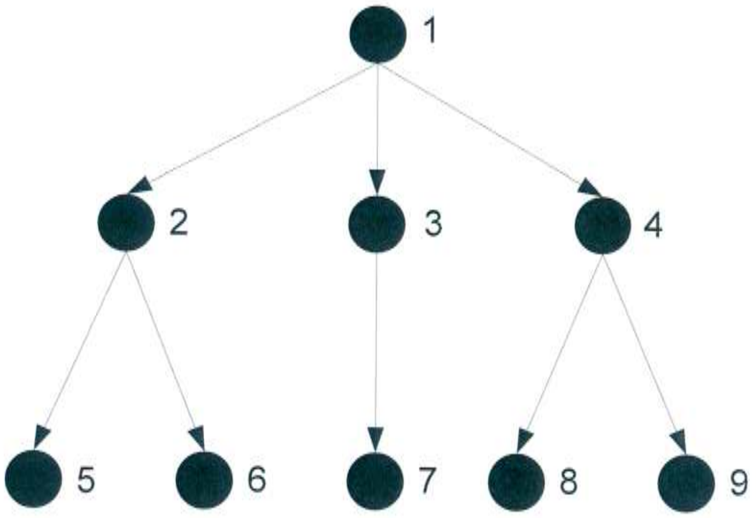


Figure 3.5.: Exemplary social network to study the result of different definitions of equivalence. Taken from [38]

3.8.4. REGE Algorithm

REGE is a classical algorithm designed to identify maximum regular equivalence classes in a social network, which can then be interpreted as roles. It was first described in [40], the following description is based on [38]. As the definition of regular equivalence is still very strict for application to real-world data, the REGE algorithm calculates a score that describes how close two nodes are to being equivalent. A score of one denotes a perfect equivalence and the similarity decreases while the score approaches zero. REGE is an iterative algorithm that calculates in iteration $t+1$ the scores M_{ij}^{t+1}, which measure the tentative degree of regular equivalence between nodes i and j. This value represents how well the neighbors of i can be matched by those of j. Each iteration does the following computation: $M_{ij,km}^{t+1} = \min x_{ik}, x_{jm} + \min x_{ki}, x_{mj}$ describes how well a tie from and to node i to and from a node k can be matched by ties from and to j to and from m. This value is weighed with the tentative regular equivalence score M_{km}^{t}, as k and m might not be perfectly equivalent. The algorithm then iterates over all neighbors k of node i, always taking the best matching node m from the neighborhood of j. The score is computed for every pair (i, j) and then normalized to complete iteration $t + 1$.

$$M_{ij}^{t+1} = \frac{\sum_{k=1}^{n} \max_{m=1}^{n} M_{km}^{t}(M_{ij,km}^{t} + M_{ji,km}^{t})}{\sum_{k=1}^{n} \max_{m}^{*}(\mathrm{Max}_{ij,km} + \mathrm{Max}_{ji,km})} \tag{3.2}$$

The result of the REGE algorithm is an $n \times n$ matrix filled with similarity values. These values can now be used for any unsupervised clustering technique. The UCINET toolkit we use employs bottom-up-clustering to group the nodes in a dendogram (3.6). Each node can be interpreted as a role, where every child node represents a specialization.

3.8.5. Prestige

Prestige is a measure for how relevant a person is within the social network. There are multiple possible ways of defining relevance. For the IslEnquirer system we are using the following definition [38]: A person is relevant if it has strong ties to other relevant persons. If we measure the prestige of a person i as p_i, we have the equation $p_i = \sum_{k=1}^{n} a_{ik} \cdot p_k$. Defining P as the prestige vector containing all p_i, we get: $AP = P$. This equation shows that the prestige vector can be determined by finding eigenvectors of A which belong to the eigenvalue one. Prestige will later be a factor for role estimation.

3.8.6. Term Frequency Inverse Document Frequency

Term frequency - inverse document frequency (TF-IDF) is a well-known measure developed for information retrieval applications that expresses how relevant a term is for a given document. The score $\mathrm{tfidf}(d, w)$ for a document d (represented as a multiset of words) and a word w is defined as the product of two scores: The term frequency is the relative frequency of w within d, while the inverse document frequency is the ratio of all documents to the number of documents containing w. Informally speaking, the TF-IDF score values terms that appear frequently in the document but which are rare among all documents (to exclude function words and other common terms). Note that the TF-IDF score regards the document as a bag of words, paying no respect to word order or position.

$$\mathrm{tf}(d, w) = \frac{n_{d,w}}{\sum_k n_{d,k}} \tag{3.3}$$

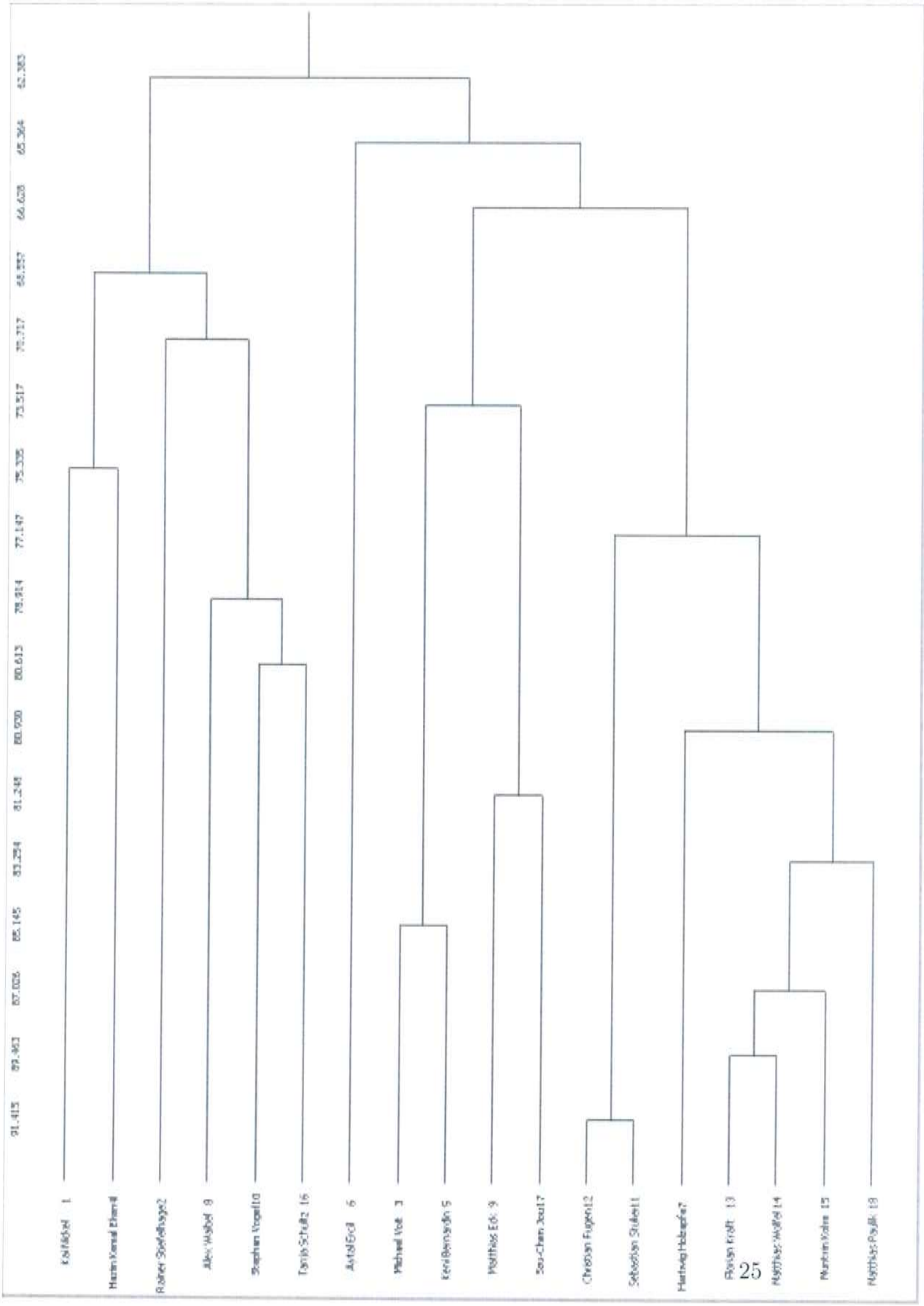


Figure 3.6.: A role dendogram created using the UCINET toolkit

$$\mathrm{tfidf}(d, w) = \mathrm{tf}(d, w) \cdot \log \frac{|D|}{|\{\tilde{d} \in D | w \in d\}|} \tag{3.4}$$

The formulas 3.3 and 3.4 give the mathematical definition of TF-IDF where $n_{d,w}$ denotes the number of appearances of word w in document d and D is the set of all documents.

TF-IDF score vectors can also be used to determine a similarity between two documents. This is usually done by calculating the cosine similarity which depends on the angle between both vectors in the space spanned by all terms. For two vectors A and B of TF-IDF scores with respect to the same set of terms, the cosine similarity is defined as:

$$\Theta(A, B) = \mathrm{acos}(\frac{A \cdot B}{|A| \cdot |B|}) \tag{3.5}$$

3.8.7. Precision & Recall

Precision and recall are two quality measures we will later use to evaluate our results. Given is a set of data points S and a subset $P \subset S$ of positive examples, i.e. data points satisfying some condition. P is unknown to the algorithm which tries to estimate it. We call this estimation $\hat{P} \subset S$. Precision now measures how many data points estimated positive by the algorithm actually are in P. Recall on the other hand describes how many positive examples are contained in the estimation $\hat{P}$. Usually, there is a tradeoff between both quantities, forcing the developer to decide which one is more important for the application. The formal definition of both quantities:

$$\mathrm{precision}(P, \hat{P}) = \frac{|P \cap \hat{P}|}{|P|} \tag{3.6}$$

$$\mathrm{recall}(P, \hat{P}) = \frac{|P \cap \hat{P}|}{|\hat{P}|} \tag{3.7}$$

4. Data Representation

We motivate a non-trivial data representation for our social user models by the fact that all data acquired during the built-up of these models is prone to noise: ASR and NLU are imperfect, people provide wrong or outdated information, users misunderstand the questions of the system, different people speak differently of the same topic. The architecture of *attribute containers* described in the following accounts for this noise.

4.1. Person Entry

The goal of the IslEnquirer is to build social user models, so the design of the data representation centers around persons. Each person is represented by a *person entry* object. The main goal of this structure is to provide a collection of social attributes described in greater detail in section 4.2. Alongside the attributes, other information concerning the dialog is stored in these objects:

- **Name** the full name of each person is required as a key to the database and to personalize the questions according to the person's knowledge.
- **Interest** a real value ranging from 0 to 1, which describes how interesting the person is for the system. As the IslEnquirer has the goal to collect data on the institute members, this value is high for core members, medium for associated guests and students and low for all other registered people. The value is first estimated during the offline step and then subject to change during the dialog.
- **Reliability** this information tracks how reliable the information provided by this person is, i.e. the value increases when questions are completed successfully and decreases otherwise. This value accounts for the fact that some people are more familiar with this or similar systems and the data extracted during their sessions usually

is of higher quality. Reliability influences the impact of this person's updates during dialog, cf. 4.3.

- **times seen, times mentioned** store how often the user already talked to the system and how often his name was mentioned by other people. This information influences the length and style of dialog, cf. 8.2.4.

4.2. Attribute Container

An *attribute type* is a feature of a user model we are interested in. The IslEnquirer handles the following attribute types, specialized for the academic domain it is designed for:

- **Importance** reflects for every other person in the database, how important this person is for the attribute owner and how strong the tie between both persons is.

- **Research group** describes the research groups people are in. This can refer to a group of specialists for a shared research area or an interdisciplinary group working on a common goal.

- **Role** is the role a person occupies within the institute. We make no assumptions on the content of this attribute as based on a prestudy (cf. 7.1), we expect very varied answers for this attribute type. The offline initialization – which requires some assumptions to produce hypotheses – reflects roles representing hierarchical position and official rank or title.

- **Research interest** is the general subject the user is currently working on. In contrast to groups and roles, research interest is not shared among different persons as it is a description of a person's individual occupation and more specific than e.g. research groups.

- **Project** contains official projects with a specified goal and a clearly defined group of participants.

- **Mood** describes a person's general feeling towards its work, e.g. whether he is optimistic to make progress or is depressed by several drawbacks. This attribute was introduced to study the behaviour of the system concerning more dynamic attributes.

The simplest way of representing attributes would be a simple variable storing a descriptive string. This design has some major drawbacks: we expect to get different answers describing the same attribute in the real world. For example, one person mentioning a "machine translation" group and another describing a "speech translation" group might refer to the same group focused on translation. Different viewpoints made them choose different labels for the same concept. This shows the necessity of having multiple labels for each entity. With that extension also comes the need to score the different labels to distinguish them in quality and frequency of use.
For many attributes, it is difficult to give a clear assignment of one attribute value to each person: For example, the offline step will generate multiple research group hypotheses for each person. Simply choosing the best alternative will remove the opportunity to later change to a then more promising attribute value. Again, this implies the use of multiple weighted attributes for each person.

The design of the IslEnquirer accounts for these observations. For each attribute type, every person is connected to its own *attribute container*. These store weighted links to one or more attributes that can be shared among multiple containers. The weights are also called *association scores*. Additionally, every attribute contains a weighted list of labels. This generic structure allows the designer to quickly add new attributes to the system.

The association score is not only important to determine the most likely attribute in a container, but is also used to calculate a *confidence* for each container. A confidence score helps the system to identify attribute containers that require further updates. The confidence score used in the IslEnquirer is based on two parameters: Firstly, it is inversely proportional to the entropy of the normalized association scores interpreted as probability distribution. Secondly, the confidence is based on the number of times the container was updated. This is represented by a parameter called *success score*. Every time the attribute container is successfully updated, this score is increased by multiplication with a fixed factor. Aborts and rejections result in attenuation of the success score. Both partial scores are multiplied to calculate the final confidence. Formula 4.1 gives a mathematical definition of the confidence, where $\mathrm{score}(a, A)$ is the normalized association score of attribute a in attribute container A and s_A is the success score of A.

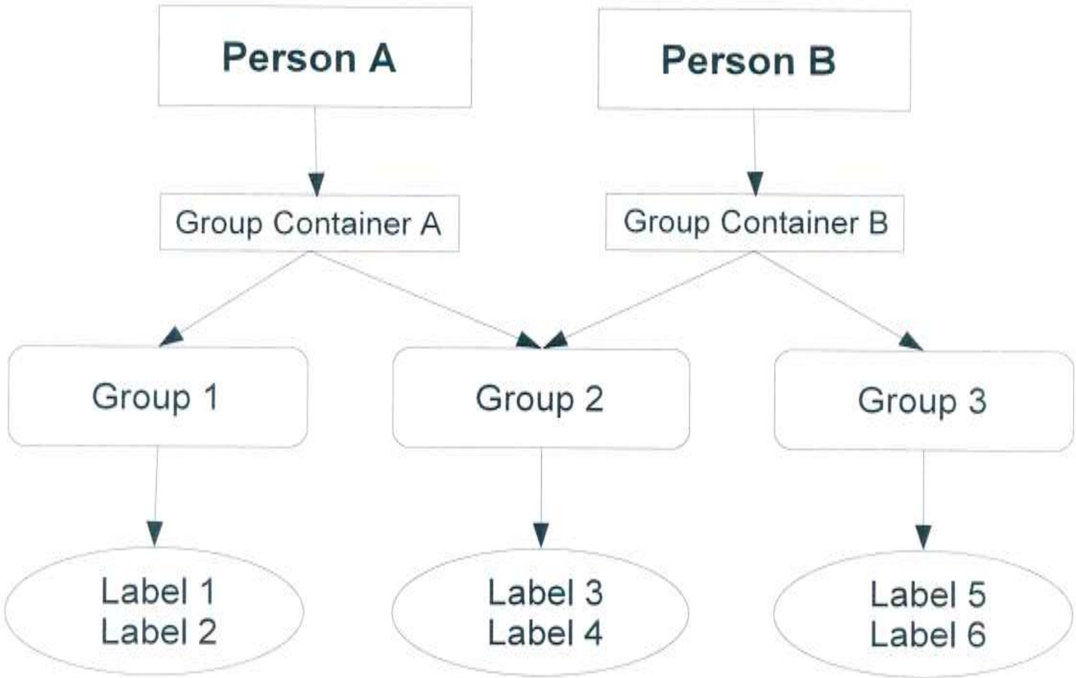


Figure 4.1.: An generic example of attribute containers, attributes and labels

$$\text{confidence}(A) = \frac{-\sum_{a \in A} \text{score}(a, A) \log(\text{score}(a, A))}{\log_2(|A|)} \cdot s_A \tag{4.1}$$

To guarantee the validity of the confidence score (given no systematic errors like consistent lying), we make two assumptions: stability and uniqueness. Uniqueness means that for each attribute container, there is one primary target attribute that describes the user's real attribute value. This is the configuration with the smallest entropy and our confidence definition prefers containers approaching it. For most attributes, this is a valid assumption, as most institute members will concentrate on one research group and one project. But it is important to note that the confidence undergoes graceful degradation if the assumption is relaxed: If the attribute score is high for several entries but close to zero for most other entries, the entropy will still be low compared to a container in which all attributes have a significant association score. Additionally, if multiple updates create the case of ambiguous entries, the success score will have increased and therefore the overall confidence, too.
The other assumption is stability. This means that changes in attributes

have to be slow enough to allow several updates between two changes. Slow, gradual changes constitute no problem for the system as they will come up in answers over time and will rule out outdated information. If the changes become to frequent, entropy will increase and no clear best attribute value can be identified.
To avoid that after a certain number of updates all attribute scores are settled (because the success score is high enough), the success score is gradually decreased if the last update is too old. This will continue until another update resets the attribute age again. The maximum age is attribute dependent to allow for different paces of change: While research interest will change slowly over time, a user's mood can change almost daily.

The IslEnquirer offers the possibility to assign an intrinsic score to attributes, representing their overall quality independently of specific association scores to attribute containers. This is not used in the current system as we observe that each attribute is either associated with a single person (in that case, the association score already contains all information) or represents a group of people and the entirety of all association scores describes the quality of the attribute. For the first case, examples are research interest or importance connections. For the second case, examples are research groups or roles. The latter class of attribute types (called *group-like*) is defined by their constituting members. There exists no research group to which no person has a strong affiliation.
Furthermore, there is no general way of assigning the confirmation or rejection of an attribute value either to the attribute quality itself or its association score to the target person's attribute container. The system uses *existance questions* ("Is there a speech recognition group at the ISL institute?") to perform this distinction in some cases. This shows the possibility of asking for the existence of an attribute value without regarding a person's connection to it. The reverse case however, i.e. asking for affiliation to a certain attribute without implicitly asking about its quality, is not possible by any means.

4.3. Processing Updates

After we established the general social user model representation, we have to define how new user input is integrated into an existing model. Updates are triggered on multiple occasions: During the dialog (see 8.2.4.4), by feedback received from the web site (cf. 4.5) or when replaying recorded sessions (see 8.6). This sets the requirement of a generic updating procedure.

Updates are carried out on attribute containers, on labels associated with attributes and on meta data. All types of updates are covered in the following sections.

4.3.1. Attribute Update

As the social user model consists of a non-trivial data structure, the update procedure also becomes complex. Each run of the updating routine performs multiple changes:

1. Changing the association score of the target person's attribute container to one or more attributes. This includes adding new attributes to the attribute container. Additionally, this might affect other attribute containers, for example if they contain a group-like attribute involved in the current update.
2. Changing the association score of labels to attributes, also including the addition of new labels.
3. Possibly changing the composition of group-like attributes by merging or splitting.
4. Changing the success score associated with the updated attribute containers.

All this is done in one update routine. This is necessary because none of these aspects can be treated without regarding the others. Especially the usage of a label is inseparable from the attribute itself. From a modeling point of view, we would like to seperate the facts that a person is associated with a certain attribute (e.g., a research group is defined by the set of its members), and the fact that this attribute is referenced by a certain label. However, labels are the only natural way for users to reference attributes during dialog, at least if we want to hide the technical definition from the user. It is unclear how a group-like attribute defined as a set of person entries can be conveniently presented to the user and how he can easily comment on its composition via speech. When we resort to labels for attribute reference, there is no generally applicable way of to identify one entry in a set of attributes with overlapping labels. This requires the system to update all attributes that may have been referenced by a used label, be it a preselected label that was used for a confirmation question or a label extracted from a free answer. The consequence is that attribute and label update are intertwined and have to be done at the same time.

We now assume that one of the mechanisms described in 4.3 has triggered the attribute update. The routine receives as parameters: The used attribute label, a success flag indicating the confirmation or rejection of the label, the current question (containing the currently targeted person) and a parameter identifying the *channel* which triggered the update. The update procedure now executes the following algorithm:

Set the dialog result

The *dialog result* is 1 when the success flag is set and 0 otherwise. For a simple slot based knowledge representation, the dialog result would be interpreted as a boolean flag whether the extracted response should be included in the model or not. The IslEnquirer uses a more sophisticated approach, which will step by step modify the dialog result, resulting in a fuzzy association score between 0 and 1. The *final* dialog result at this point depends not only on the outcome of the dialog but also on the currently updated attribute container. It is later used as the new association score.

Determine the attribute objects eligible for update

The update routine scans a list of all attributes of a valid response type. Every attribute is now checked whether it fits the user's response. For most attributes, this simply means checking if the set of labels associated with the attribute contains the response string. For importance weight atttributes, we have also have to check that the source of the connection is correctly set to the target person. Attributes in the target's attribute container are always eligible for update.

The attribute type the user was referring to is either determined by the attribute type of the dialog question or by the formulation the user employed ("I am in the dialog *group*"). There can be multiple attribute types be involved in the updating procedure.

Determine the attribute containers eligible for update

All attribute containers containing one of the attributes determined in the previous step are now selected and updated. This is done by scanning a list of all attribute containers belonging to the type of the attribute in question. Additionally, the attribute container for the target person is always selected[1].

Calculate new attribute score

The new attribute association score is calculated for every container selected in the previous step as a repeated linear interpolation of the dialog result with the old value:

$$d_i = \alpha_i \cdot x + (1 - \alpha_i) \cdot d_{i-1} \tag{4.2}$$

where x denotes the old association score for this container. x is 0 if the attribute is not contained in the container. This can only happen to the attribute container which belongs to the current target person, as this one is always selected for update. d_i is the tentative dialog result after applying interpolation step i. d_0 is the initial dialog result as determined in the first step, d_n (were n is the number of interpolation weights) is the final dialog result and used for the updates in the next steps. The interpolating weights α_i are determined by several criteria:

1. **Confidence** The higher the confidence of the attribute container, the less is the influence of a new update to it. The reason behind this decision is, that every update to a high-confidence attribute container is either already represented by it or has a high chance of being noise. Note that to still allow attribute changes that correspond to changes in the real world, the confidence is regularly temporarily decreased. Cf. section 4.2 for details.

2. **Label** All attributes that fit the extracted response label are updated but not all attributes fit this label equally well. This is expressed in the score of the label for the attribute. Attributes which are strongly connected to the response should receive a stronger update than those for which the reponse is found at the bottom of the label list. We therefore use the inverted label score as another interpolation weight.

[1]with the exception of existence questions, cf. section 8.2.1

3. **Importance** The IslEnquirer does not only contain questions about the user himself but also poses questions that target other people. This information may not be as reliable as information given directly on the user. Furthermore, the quality depends on how well the user knows the target. In our system, this is expressed by the importance connection score from the user to the target. This value is directly used as an interpolation weight.

4. **Channel quality** The update procedure can be used for several different input channels: from dialog utterances parsed through the context free grammar, detected in the output of the n-gram speech recognizer, or derived from a click on a feedback button on the web site. Each channel has its own chance of misrecognition, which is reflected in different channel interpolation weights. For internet updates, this weight is set to zero, which results in ignoring the old value in this iteration. Updates from the n-gram channel have a weight close to one, meaning that the dialog result will have little influence on the new value.

5. **User reliability** The system will encounter users who are trained in using spoken dialog systems and are willing to cooperate. On the other hand, there are untrained and uncooperative users, whose responses should not get the same influence as they are more error prone. Therefore, the system keeps track of a reliability score for each person entry, which can directly be used as an interpolation weight. The reliability scores are updated after every question subdialog, cf. 4.3.3.

6. **Question reliability** Having variable reliability is also true for different types of questions. We assume that open questions have a higher chance of misunderstanding and their results therefore a lower reliability. But as we noticed during the prestudy and testing, even the wording of a question can effect the quality of its answers. The IslEnquirer therefore uses a finer granularity, modeling reliability for every question separately, cf. 8.2.3. This value is then used as an interpolation weight in formula 4.2.

7. **Target** The IslEnquirer system can map answers of one type to answers of another type. This becomes necessary if a user gives a response that is of a different type than the expected response type for the answered question. The most important example for this

is an answer of type `response_group` (describing a research group) to a question expecting an answer of type `response_role`. If a question is tagged to have deviant answers, the alternative attribute containers are also updated. As this conversion is only based on a heuristic, we introduce a penalty factor to reduce the effect of the update.

8. **Indirect update** Each update is not only performed for the current target's attribute container, but also for all other containers of the same type which contain an attribute fitting to the response. The reason behind this is that the usage of a label to describe a concept is an indication that attribute hypotheses having it attached are valid and useful descriptions of the real world. Since this is again just a heuristic, there is another penalty for this kind of update to avoid making momentous mistakes.

Update and insert attributes

After the final dialog result has been determined, the respective attribute containers are updated and the attribute association scores are changed to their new value. The container associated with the current target person is always updated with each fitting attribute, even if it does not contain it. In this case, instead of changing the attribute weight, the attribute is inserted as a new hypothesis.

Attribute insertion

If no attribute fitted the reponse string and the container associated with the target does not contain any attributes, no update has been performed up to this point. To be able to represent the response in the social user model, we create a new attribute and insert it into the empty container.

Label update

In addition to changing the score of the attribute itself, we also have to update the label scores to account for either having used this label successfully (or unsuccessful, if the question was aborted) in a confirmation question or for a user uttering this label to formulate his response to an open question. Every attribute that fits the extracted response will receive a label update. This follows the same procedure as the attribute update, using the same nested interpolations. Instead of the attribute container confidence, the label score is used.

Confidence update

Section 4.2 describes how confidence is calculated for attribute containers. The entropy factor of the confidence can be directly derived from the container itself and thus was already updated implicitly. It remains to update the success score, which reflects the recent sequence of successful and unsuccessful updates for each attribute container. The new value is calculated according to one of the following formulas, depending on the outcome of the completed question:

$$\text{successScore}_{\text{new}} = \text{successScore}_{\text{old}} + (1 - \text{successScore}_{\text{old}}) \cdot m_{inc} \tag{4.3}$$

$$\text{successScore}_{\text{new}} = \text{successScore}_{\text{old}} - \text{successScore}_{\text{old}} \cdot m_{dec} \tag{4.4}$$

where m_{inc} and m_{dec} describe the factors by which the distance of the success score to 1 (representing perfect confidence) is increased respectively decreased.

Example selection

As described in section 8.2.4.2, the system uses attribute-label pairs with the highest combined score as examples to give the users guidance in what answers to an open questions it expects and can understand. As scores have changed after the update, the example list is updated at this point.

4.3.2. Updating group-like attributes

In this section, the term *group* is used to refer to any group-like attribute, for example a research group or role.

Up to this point, the only ways that allow group-like attributes to change are the manipulation of association scores to existing attributes or the addition of new members to an existing group. This may be not sufficient. The offline hypotheses might not contain the groups required to describe the world accurately, thus the system needs the possibility to create new groups. We implemented two ways to do so: Two existing groups can be

```
method update(response):

  if response was confirmed:
    dialogResult = 1
  else:
    dialogResult = 0

  for all attributes of valid type:
    if attribute fits response:
      for all attribute containers of adequate type:
        if attribute container contains attribute:
          for all interpolation weights alpha:
            dialogResult = alpha * oldAttributeConnectionWeight + (1 - alpha) * dialogResult
          update attribute container with dialogResult
      if dialogResult == 1
        dialogLabelResult = 1
        for all interpolation weights alpha:
          dialogLabelResult = alpha * oldAttributeConnectionWeight + (1 - alpha) * dialogLabelResult
        update attribute label with response, dialogLabelResult
```

Figure 4.2.: Simplified version of the updating procedure in pseudo code

merged and an existing group can be split up. The decision is based on a quality score described later. This check is performed when an attribute is updated or (in the case of research groups) when importance connections are updated. The latter is due to the fact that group affiliation is correlated to and based on (during the offline computation) personal relations.

4.3.2.1. Splitting a Group

To split a group, we have to distribute all concerned persons to two new groups. For a group with n members, there are 2^n possible splits and this may be too large for exhaustive online search on large groups. Instead, the system tries to find an optimal splitting and compare the resulting groups via a quality measure $\hat{q}$ to the existing group. The algorithm starts with two seeds, which are the nodes with the largest distance. This distance value is based on two factors: Firstly, the system calculates the relative overlap of labels employed by users to reference the group. The larger the overlap, the smaller the distance. For research groups, we also factor in the importance weight between the two nodes in both directions. When the seeds are found, all nodes are assigned one by one to one of the two tentative groups, depending on the averaged distance to all nodes in them. $\hat{q}$ measures the averaged distances between all node pairs in a group, weighting each pair by the interest score of both nodes. If the new groups yield a better $\hat{q}$ score, we use them to replace the old entry. The association score for each member to the new group is the same as the association score for the old group multiplied by the relative improvement in $\hat{q}$ gained through splitting. For label selection, we merge all labels used by its members.

4.3.2.2. Merging groups

To decide whether the fusion of two groups is beneficial, we simply have to merge the two sets of person entries and compare the respective $\hat{q}$ scores. If the score of the merged group is better than the scores of both existing groups, we replace the old entries by their union. This is done in the same way as described for splitting groups. As merging is less complicated and more robust than splitting (where we do not evaluate all possibilities), it is desirable to tune the offline initialization in a way that forces the creation of many small, reliable groups. This increases the chance that all groups in the real world can be represented as unions of already existing groups. Independently of quality aspects, groups also have to be merged when they are very similar in their composition to reduce redundancy.

4.3.3. Meta Data Updates

Meta data describes information that is not part of the social user model itself but is important for its acquisition. In contrast to the attribute update, some of these parameters depend on the mode of update. During a dialog session, meta data update is triggered immediately after the attribute update.

Question reliability

Question reliability is one criterion for question selection during dialog and also influences the update procedure. It is updated for the current question on completion. If the question was aborted (e.g. for taking too many turns), the reliability is decreased. Else, it is increased by a factor determined by the length of the dialog (the shorter the better).

Frequency counters

Regardless of the outcome of the question, the system increases the frequency counters for the current user-question and user-topic pair. These are used as a criterion for question selection (section 8.2.3) and usually suppress the repeated use of one question within a short period of time.

Satisfied targets & last target

To control the dialog flow, attribute types that received an update with dialog result > 0 are collected in a set of satisfied targets for the duration of one session. Additionally, the last targeted attribute type is stored. Section 8.2.3 describes how these parameters influence the question choice.

Person interest

As noted in section 4.1, each person has an interest score denoting how interesting it is for the task of collecting information on current institute members. This score is calculated during the offline step and updated during the new user registration, but can also be changed in several ways while answering questions: there is a dedicated relevance question, which asks the user whether another entry in the database of the system is an active member. Furthermore, if a name is mentioned as a response to a regular question (usually, an importance question), this person's interest is updated. Every way of updating the interest uses the same method, increasing or decreasing the interest value by a certain percentage, which is determined by whether it is a direct update (e.g. answer to a relevance question) or an indirect update (usage of a name while answering an importance question).

4.4. Handling Uncertainty

We have to expect that not all answers, even if they are understood and interpreted correctly, are useful for the system. Some users will probably provide information of wrong granularity, uninteresting information, wrong or outdated information. This will be especially the case for open questions and questions on other users. In the following, we will shortly recap how the techniques described in this and other chapters will mitigate this effect:

- Each attribute container maintains multiple hypotheses in parallel. If one attribute is rejected, others can replace it. In return, if errors or misunderstandings inject erroneous information, it does not overwrite the existing data which can be recovered later.
- If inconsistent information slips into the database, it will lead to a high entropy caused by contradicting attribute values. This quickly

leads to a high priority of the associated attribute container during dialog question selection and will bring up questions to solve the tension.

- The IslEnquirer does not rely on a single source for gathering information. The offline step already creates a preliminary social user model which is then validated and complemented by information gathered during dialog. Multiple participants contribute data for the same attributes, either via third person questions or indirect updates.

- By giving examples and making use of confirmation and reconfirmation questions, the system tries to guide the users in what level of granularity and detail the system expects. In many cases, this also increases the fraction of parsable user inputs which in return leads to fewer recognition errors.

4.5. Web Site Presentation

The goal of the IslEnquirer is to gather information about its users and to present this information on an automatically generated web site. The site is dynamically created using a PHP script. The usage of a MySQL server as a database makes it easy to select, order and filter the stored data we want to present.

This web site is structured as follows: The index page contains an overview of all registered institute members whose interest score exceeds a certain threshold. To further distinguish between more and less relevant entries, less interesting people are greyed out. Clicking on any entry will lead the visitor to a detailed page on the selected institute member. The page gives the member's name, shows a photo and presents the member's most important contacts, his research interests, the research groups he is in and his roles in the institute. The photo is automatically extracted from the data recorded during the dialog sessions. We chose the image associated with the highest face recognition score as we assume this is correlated with a representative face image. The attributes are labeled with the terms acquired during offline data collection and dialog. The entries are sorted by their score to list the best hypotheses first and so are the labels describing them. Clicking on one of the entries leads to another overview page showing all other people associated with this group, role, etc.

All entries have two buttons to give simple feedback: "thump up" and "thump down". Pushing these buttons has two effects: on the one hand, the negative or positive feedback is counted and stored in the database. This allows simple evaluation of the whole database outside of dialog interaction or questionnaires. On the other hand, this feedback can be used to update the database itself, as the update formula is generic enough to handle input from different sources. Only the channel parameter, that determines the noise level of the input channel (zero for updates over the web site) has to be set. To make the update more reliable, the user can select his name from a given list generated from the database. This allows to weight those updates stronger that deal with attributes the user is familiar with. Note, that while this update requires the user's initiative, it requires minimal effort and no typing at all in contrast to manual database editing. Furthermore, the user does not have to know the "correct" values for wrongly filled fields. Instead, he can just press the button for negative feedback to decrease the score of a misplaced value and it becomes replaced by another hypothesis. If none is available, this will force the dialog to bring this question up as soon as it gets the possibility to do so.

4.6. Modularity & Extensibility

The IslEnquirer attribute and question architecture is designed to quickly integrate new entries. This allows the flexible adjustment of the IslEnquirer to new tasks and user groups.

For inserting a new dialog question to an existing attribute, only the new system utterances must be entered in the utterance file and one line has to be added to the database. The latter contains the question type (e.g. open question or confirmation question) and other parameters. This minimal effort makes it very convenient to add new questions, which on the one hand keeps the dialog new and interesting for expert users and on the other hand allows to easily experiment with multiple question sets to test the influence of different formulations and question types. The new question is then automatically integrated at startup by the question generation (cf. 8.2.2).

Adding a new Attribute requires only the implementation of the abstract classes `AttributeContainer` and `Attribute`, from which most methods

Personal Information on Hartwig Holzapfel

His/her research interests:

dialog	0.811673271032306		
dialogue	0.355752103314392		
intention	0.189263402674007		
speech	0.165266991024636		
grammar	0.157365287041249		
dialogue systems	0.15135795146783		
robots	0.148987604619698		
speech recognition	0.146798286838193		
user	0.141926287419822		
persons	0.141712912872828		

Figure 4.3.: A screenshot of the dynamic web site, showing parts of a social user model.

can be reused. Especially update processing and confidence calculation are independent of the attribute type. The new attribute then has to be registered for the person entries and in the database extraction method. For useful integration, the designer must of course provide new dialog questions associated with the attribute.

To achieve a full transformation to another domain or scope, the main work has to be invested in the offline step. Appropriate data has to be collected, a new vocabulary must be generated and for completely new attributes, new algorithms for hypotheses generation should be developed.

5. System Architecture

While chapter 3.1 gave a generic overview of the typical building blocks of a modern spoken dialog system, we will now shortly review the actual components and preceding work. Figure 5.1 shows all main components which are part of the IslEnquirer. The bottom three components in the box marked "IslEnquirer" form the dialog manager that was developed for this setup. It is based on experience and code from earlier experiments in a hallway scenario, namely the receptionist experiment from [20, 19]. The offline processing component itself does not belong to the dialog manager but uses a publication data corpus to determine initial hypotheses on the social user models. These are passed through a database to the dialog component.
For speech recognition, we rely on the Janus using the IBIS decoder [34] with a context free grammar language model, with built-in support for OOV detection and two-phase recognition for names. This is complemented by a second Janus version using a n-gram language model. The NLU and basic dialog processing is done by Tapas [18], the user identification component was developed in [14].

5.1. Modular Structure

The IslEnquirer dialog system consists of multiple dialog modules. Each module focuses on one self-contained part of the dialog. Transitions between modules are modeled by a finite state machine and handled by a global dialog strategy. The latter also calls initialization and clean-up routines on entering and leaving a module. Figure 5.2 shows all dialog modules of the IslEnquirer system. Each module contains its own routines for input interpretation and dialog move selection. This allows the strategy to react differently to input tokens based on the current module. For generic system actions (e.g. repeating the last utterance), this is complemented by a global action selection. Information is passed between different modules by using a slot model. Each module can be replaced separately without affecting other modules. In the following, we give a

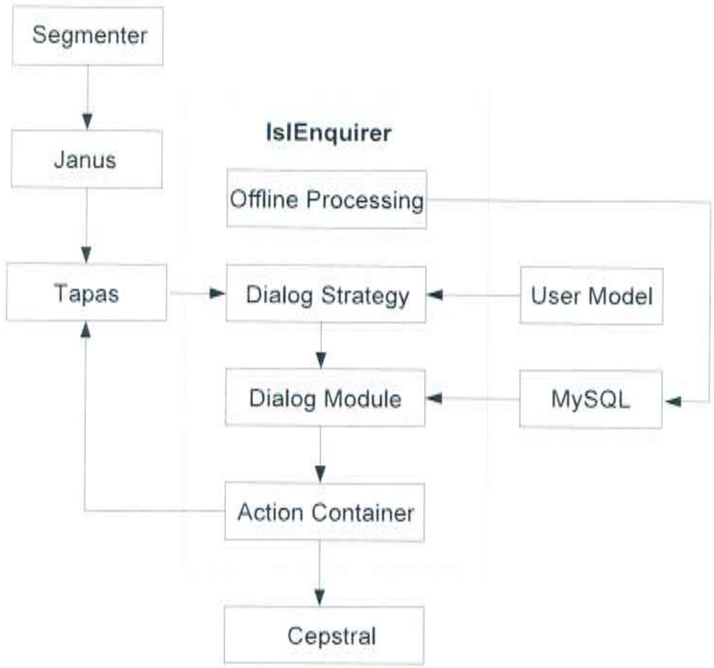


Figure 5.1.: The IslEnquirer system architecture

short description of every module, chapter 8 will describe the main parts of the system in greater detail.

- **Idle** Waiting for the next user
- **Greeting/Task Info** Tries to attract the user, greets him and explains the task
- **Name Learning** Identifies the user by asking for his name or for confirmation of a hypothesis of the user model
- **New User Registration** Gathers basic information on first time users, determines their relevance for the task
- **Social User Modeling** Poses questions to gather more information for the social user models
- **Coworker/Concept Name Learning** Learn new labels to describe attributes of the social user models

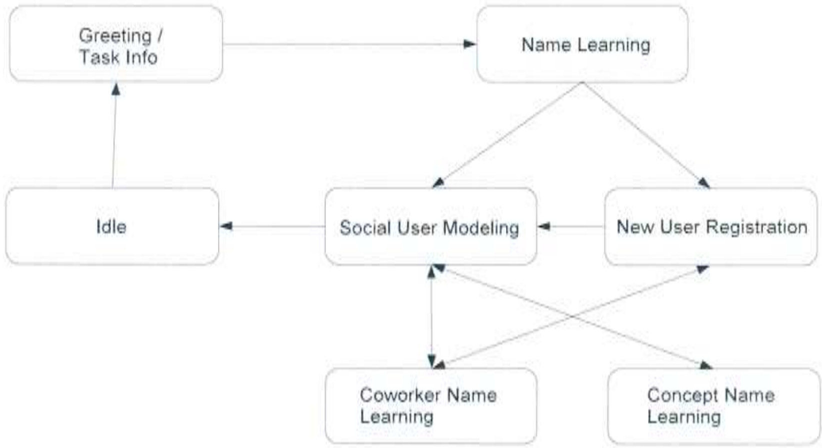


Figure 5.2.: Finite state machine describing all dialog modules and the possible transitions between them

5.2. Robust System

The goal of the IslEnquirer system is to end up with a fully autonomous robot which is able to learn from past dialog sessions. This requires to build a system which can store the gathered information in a persistent fashion. This is supported by all components of the IslEnquirer system. All information on the social user models is immediately stored in a MySQL database[1] during a session. This protects the system from data loss on planned or unintended system shutdowns. It also allows easy displaying and altering of the data using different interfaces, for example using a web interface. For more details, cf. to section 4.5. The database is also a convenient way to integrate the initial user models, which are created during the offline step, in the interactive system. Furthermore, all updates to the user models are logged in a generic file format which allows to replay selected sessions. Refer to section 8.6 for details.

Both identification components (face id and voice id) support online training. ASR and NLU allow the addition of new terms during run-

[1] `http://www.mysql.com`

time. Whenever new names or labels are learned, they are passed to both components to include them in the respective data structures.

5.2.1. Session Model

The session model handles the state of a session and the transition between sessions. This becomes important, once the system runs over a longer period of time, interacting with different users. It can send session start and stop messages to trigger recording of all perceptual input and to (de)activate user identification. The session model also contains a timer to identify unanswered turns. When this timer is triggered, the session model will continue with another turn or abort the running session when it assumes that the user has lost interest or the microphone is not working. A new session may be triggered by the user identification component when a face is recognized or speech was detected by the segmenter or by the dialog manager, when semantic input was passed to the dialog manager or the system starts speaking.

The session model also suppresses triggers which would normally start a session for a fixed period of time after the end of the each session. This avoids tracking and pestering a person that has just finished a session and is still standing in front of the robot afterwards. To display its temporal inactivity, the robot's head is placed in a resting position, looking at the floor. During this time, the session models sends a message to notify the user identification components whether the gathered data of this session is used for training or is discarded. In the case of a training request, the desired user id is sent to enforce consistent ids among all components.

When a session is completed, the session model enforces a reset of all session dependent models to clear all states and slots for the next user.

5.3. Flexible Architecture

The modular approach described in section 5.1 makes it easy for the developer to change individual dialog modules for modified experiments: only the module itself and the transitions need to be modified. This has been successfully carried out by using a different name learning and question answering module for a different experiment. Furthermore, Tapas supports the easy replacement of the global dialog strategy. We use this mechanism to introduce a new modular approach for Wizard of Oz (WOZ) experiments. This is an experiment for testing a dialog system and gathering data without a running fully autonomous system: the robot has the recognition and acting capabilities of the final system but is controlled by a human "wizard". The user is told that he is interacting with an autonomous system [6].

In addition to the autonomous strategy, we implemented a `WozStrategy`. This implements the dialog strategy interface and comes with a graphical user interface which allows the operator to change between modules and select actions. To minimize code duplication, all activities the strategy can perform (triggering dialog moves, database access, data structure manipulation, ...) are encapsulated in an *action container*. The buttons activated via the WOZ user interface and the acts chosen by the autonomous strategy call the same actions in the action container now. The update routines to process user input are implemented within the autonomous dialog strategy only and referenced there from the `WozStrategy`. This enables the developer to carry out WOZ experiments which offer the human wizard the identical set of moves that is also available to the autonomous strategy.

Via this architecture, any strategy which implements the lightweight strategy interface can replace the current ones. This could be helpful if other transitions between modules instead of the current finite state machine are desired.

6. Offline Network Analysis

6.1. Social Network Construction

In theory, the dialog component could be run without any database initialization, as it is able to learn and process new information. For some attributes, for example projects, this is in fact the case as we are currently not able to gather information on this topic beforehand. There are however several problems with this approach: A system without a priori knowledge would have to resort to only open questions and is not able to assist to the user by giving examples or by using confirmation questions. Our prestudy suggests that open questions are harder to answer than closed ones with only a limited set of possible answers. In addition, ASR accuracy and grammar coverage will drop for open questions and without any preparation we can only guess what the important vocabulary might be.

Therefore, we initialize the database with as much information as possible while still keeping a fully automatic system which does all the information retrieval and extraction without human pre- or postprocessing. To this end, we extract information from a publication corpus which is manually gathered from the official ISL web site. In the future, we will crawl the internet for information on known people at the lab, using the iFinder [17] system. Each entry of this corpus contains the year, the title and the authors of one publication. Where available, we also extract the abstract of the publication from an attached PDF document[1]. To exclude wrongly encoded documents or documents in other languages, we employ a simple classifier which counts the appearance of the most frequent English words in each document and checks this number against a threshold. From the gathered data, we try to identify the current institute members, estimate their attribute values and extract labels describing these attributes. Most of this process relies on social network analysis.

[1] `http://www.pdfbox.org`

Before we built a social network, we select the people which we are interested in, i.e. the members of the ISL institute. For this purpose, we calculate the initial interest score, which we need anyway for initializing person entries (cf. section 4.1). Entries which go below a certain interest threshold are excluded. This has the benefit of less noise in the initial database and more focused attribute and label hypotheses. The data we use for the offline step is extracted from the official web sites. We can therefore assume that current institute members will appear more frequently and more recently in the list of publications than former members or associated researchers. This leads to formula 6.1 for estimating the interest value, where $\tilde{P}$ denotes a desired number of publications, p_v the total number of publications by person v, a_v the age of the last publication by person v and $\tilde{A}$ a desired maximum age of the last publication.

$$\text{interest}(v) = \frac{1}{1 + \max(0, \tilde{P} - p_v)} \cdot \frac{1}{1 + \max(0, \tilde{A} - a_v)} \tag{6.1}$$

We then use the collected data to build a social network. A social network is a (directed) graph, where every node represents one person in the database. The nodes are connected by weighted edges. The weight of an edge depends on the frequency of joint publications. To account for changes in affiliations and collaboration schemes over time, each joint work is weighted by its age so that recent publications have a higher impact on the edge weight. The exact formula is given in 6.2, where $\text{Pub}(A, B)$ denotes the set of all publications by both A and B and age_p is the age in years of publication p.

$$\text{importance}(A, B) = \sum_{p \in \text{Pub}(A,B)} \frac{1}{\text{age}_p + 1} \tag{6.2}$$

Weights are then normalized to sum up to one over all outgoing edges for one node. This makes it possible to interpret the weights as probabilities and allows us to calculate the entropy of the distribution associated with each node. A high entropy is an indicator for a person with many different co-authors and a high reputation. Low entropy indicates a person with a dedicated contact person, for example a student writing his diploma thesis. Following this observation, node entropy can later be used as a feature for role detection.

As the edges are directed and normalized, we can expect to have different weights on both sides of a link between two nodes. We can interpret the weight of the edge from A to B as the importance of B for A. An example is the connection of a student to his supervising professor. While the professor is very important for the student (with an edge weight close to one), the student is not as important for the professor (who will have many other outgoing edges and therefore a much lower edge weight).

These importance values are already the first attribute that is inserted in the database, used in dialog and presented on the web site. We associate a confidence score to each connection proportional to the weighted number of publications of both nodes. The network built in this step will further be useful to derive additional attribute types, like groups or roles.

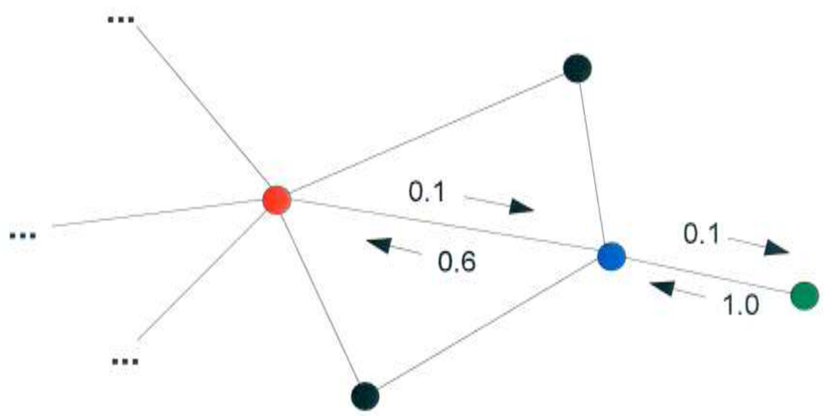


Figure 6.1.: Exemplary social network showing a professor (red), a researcher (blue) and a student (green)

6.2. Groups

Groups are a classical higher-order structure that can be identified in social networks. A group is defined in [38] as a subgraph with certain properties: mutuality of ties, closeness or reachability of subgroup members and frequent ties among members. In our understanding of the term "group", interpreted in the lab scenario, it stands for a selection of people working together as a team on a common topic. Note that common research interests are not necessary for forming a group. Take for example a group of speech, vision and dialog experts working on a joint pointing gesture interpreting system.

A usual way of finding groups in a social network is by searching for cliques, as described in section 3.8.2. It is easy to see that a clique fulfills the definition of a group. Finding cliques solves the problem of finding groups in theory but there are several problems to come by: Firstly, we identify too many cliques with a high degree of overlapping, as the removal of one node v in a clique can allow the addition of a new node w that only lacked the connection to v. To avoid having too many hypotheses, we have to get rid of near-duplicates. Secondly, the definition of a clique might be to restrictive for finding all relevant groups as not all members must have pairwise joint publications. Thirdly, the definition of cliques does not take into account the weight of ties among its members and other factors which define the quality of a group.

There were designed alternatives to cope with the problems of cliques in social network analysis and graph theory. Some of them are sketched in section 3.8.2. However, n-cliques with $n > 1$ are not suitable for the IslEnquirer scenario as all nodes are connected by a path of length at most two, which represent professors or group leaders. The k-plex performs better in this regard, but it still does not respect the quality of the created groups.

To solve all of these problems at once, we apply the following algorithm:

1. Identify all cliques in the graph induced by the social network
2. Define a quality measure $q(G)$ for groups based on their internal coherence and other features
3. Define a similarity measure $s(G_1, G_2)$ for groups which compares groups in regard of their composition and their associated labels
4. Shrinking: For every group in the tentative set, remove nodes, one in each iteration, as long as this increases the q-score of the remaining group. Always remove the node with the lowest total importance for all other nodes in the group. This step removes weak links from the groups. Simple addition of single members is easy during the dialog, so any errors made in this step can be easily repaired.
5. Merging: Get the two most similar groups from the tentative set. If their similarity exceeds a certain threshold, remove them from the set and replace them by their union. Repeat this step until no two groups are similar enough.
6. Removing: We continue with picking the pair of most similar groups but instead of merging, we simply delete the group for which q assigns the smaller score. Again, this is one until a fixed similarity threshold is reached.

Definition of $q(G)$: The quality of a group G depends on several criteria which are represented by scores in $[0, 1]$ and are integrated by multiplication. The first criterion is a high degree of connectivity among its members as we assume that normally, members of the same research group work and publish together. To this end, we calculate the normalized sum of the importance weights between all pairs of group members. The addends are weighted by the interest score of the end nodes, to suppress the influence of persons with a weak affiliation to the institute. The second criterion is research interest similarity. As stated before, we assume that in most groups, people have some overlap in research interest. Section 3.8.6 shows how a similarity of terms describing this research interest can be derived. Again, we sum the weighted pairwise inverted similarity scores to receive a score for group evaluation. The third criterion is simply the inverse of the difference between group size and a desired optimum. This score was included to make use of a-priori knowledge on the typical size a research group has. The fourth criterion is defined by the sum of interest scores of all group members. This is motivated by the requirement of an active research group to have at least some active members.

Definition of $s(G_1, G_2)$: As for the quality measure, the similarity between two groups G_1 and G_2 is calculated as the product of independent similarity scores. The first score is the degree of overlap, formally: the ratio of the size of the intersection of the two groups to the size of their union. The second score deals with the nodes contained in the symmetric difference of both groups: for G_1, we calculate the sum of importance connection scores from all nodes in $G_1 \setminus G_2$ to all nodes in G_2. We then do the same with switched roles of G_1 and G_2 and add both values. The last partial score is again based on keywords. With the same algorithm as for group quality, we calculate a similarity of terms that were extracted to describe the groups (cf. sections 6.4 and 3.8.6 for details). While we compare individual research labels to compute $q(G)$, we now compare group labels.

For merging two group candidates, we also tried a different decision rule: Two tentative groups were only merged when the resulting group is scored "nearly as good" as the better one of the underlying groups: $q(G_{fusion}) \geq \alpha \cdot \max(q(G_1), q(G_2))$, where G_{fusion} denotes the fusion of groups G_1 and G_2. With this approach, we expect to achieve a better group differentiation, probably at the cost of a larger group set.

6.3. Roles

The conception of the term *role* we will be using for the IslEnquirer during the offline step is based on formal roles in the academic environment. There are students (e.g. working on their diploma thesis), research assistants and PhD students, group leaders and professors. If we also regard people outside the ISL institute, we will add roles like guest, former member etc.

As our prestudy showed (cf. chapter 7) a great variance in the understanding of the role concept, we chose not to assign predefined roles. Instead, we will use an automatic approach to cluster people with similar roles. The offline step will not assign any labels to these role groups. This is postponed to the interactive learning component. This gives maximal freedom to the users while we can still provide a reasonable initialization.

To identify roles within the institute, we analyze the social network built in section 6.1. We begin by finding clusters of regular equivalent actors and refine this partition while evaluating other criteria. The REGE algorithm (cf. section 3.8.4) returns a clustering based on a good approximation of regular equivalence. The IslEnquirer uses the REGE implementation of UCINET, a general purpose package for the analysis of social network data[2], which allows computation and visualization of many popular algorithms.

However, we found that this clustering algorithm offered little flexibility in tuning the granularity of the partition. Social network theory offers other measures for calculating role similarity (e.g. prestige). For the IslEnquirer, we therefore introduce a generic framework to calculate roles based on several criteria, among which the REGE score is an important representative: To measure the role similarity between two people, we calculate an euclidian distance d_{role} in a multidimensional space. Every feature we want to use is mapped or normalized to a real number in $[0, 1]$. Features that were tried are:

- REGE score (inverted and normalized REGE similarity value)
- Prestige (cf. section 3.8.5)
- Number of publications to measure the length and intensity of the person's work at the institute

[2] `http://www.analytictech.com/ucinet/ucinet.htm`

- Entropy on importance connection distribution measures whether the person's relations focus on a small group of coworkers or are scattered around the whole institute
- In- and out-degree within the social network as an alternative to entropy
- Interest score, with the goal to keep current institute members and other persons in separate roles
- Number of associated research groups, with the legitimation that influential members are usually part of multiple research groups

This approach allows easy integration of new features and flexible combination of multiple criteria.

Based on this distance measure, two ways of clustering were tried: bottom-up clustering and a connectivity based approach. Bottom-up clustering came with the difficulty of finding a good value for the fixed maximal distance that allows combining two tentative role sets. A high value will result in very small, "overfitted" clusters, whereas a small value will result in too little differentiation. In addition, this value has to be estimated every time the feature set is changed. The connectivity based approach tries to circumvent these problems and works as follows: firstly, a graph G_1 is constructed which contains an undirected edge from each node to its closest neighbor based on the distance measure d_{role}. In this graph, we search for connected components as tentative role sets. This process is repeated by constructing $G_2, \ldots, G_i$, using the the closest $2, \ldots, i$ neighbors to create edges. For the running system we have $i = 4$. For each node v, only neighbors with a distance score $\leq \beta \cdot a$ are connected to v, where a is the average over all distance values and β is a scaling factor growing linearly with the number of maximal allowed connections in the current graph. This restriction avoids forcing people with a unique role (a professor might be an example in the context of a single institute) together with only roughly similar neighbors. To avoid connected components forming long chains of nodes, we employ another check to remove those candidates: For every tentative role, a fixed percentage of nodes is temporarily removed by random selection and the remaining component checked for connectivity. This is repeated several times and the role is deleted if less than a given number of tries is successful. Additionally, the maximum distance between every pair of nodes in the network is limited.

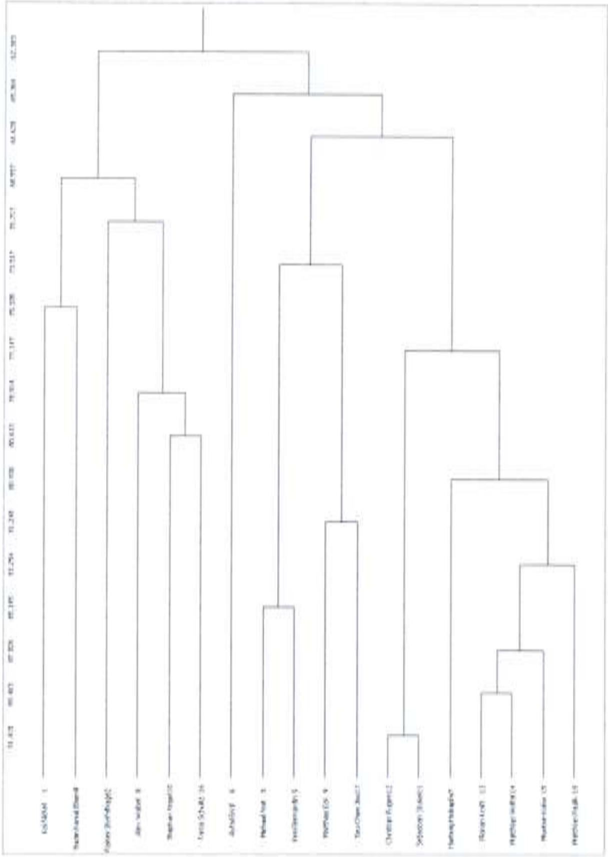


Figure 6.2.: The REGE algorithm calculates a group clustering based on regular equivalence

After the roles are created, we remove duplicate roles and associate the remaining ones to their members. It is not guaranteed that every person is contained in at least one role cluster. If none is assigned, we create a unique role especially for this person.

6.4. Research Interest & Group Labels

It is crucial for the IslEnquirer system to have a large set of words describing the concepts found in the other steps. Additionally, we need labels to describe the users' research interest. The labels are required to provide hypothesis used for confirmation questions, to form examples and to create an ASR vocabulary. Both tasks require similar techniques. We will first focus on research interest labels. To extract those, we apply a well-known technique from information retrieval: TF-IDF scores as described in section 3.8.6. In a first step, we extract a set of label candidates from all abstracts of publications which are provided to the system. This is done by gathering all words and selecting those that exceed a certain frequency threshold (to eliminate errors by the PDF text extractor that sometimes fails on words split over multiple lines), occur in a minimum number of unique documents (to exclude special terms) and that are of a useful part of speech type, i.e. a noun or a noun phrase. The latter classification is done by a Part-Of-Speech tagger[3]. As the English language uses many composite terms that consist of multiple words, we also add all bigrams that comply with these requirements.

For all persons and all terms we compute the TF-IDF score. The original definition calculates the importance of a term with respect to a single document. To transfer this to a term relevance for an author with multiple documents, we concatenate the texts extracted from his publications into a single document. Another approach which calculates the scores for each abstract individually and averages over all document scores of one author was not successful. It resulted in a collection of specialized terms scattered around publications. While still associated with the author, they did not generalize to an overall description.

As the document base is still rather small and specialized in content, we use additional documents from other sources (using data from [32]) to adjust the scores. This will reduce the score of terms that are too generic in nature but occur only in a few documents in our database due to personal wording preferences. It will also boost the score of terms that are suited as labels, but are overrepresented in documents in our publication database, preventing a high score.

The TF-IDF scores are then used to rank the terms for every person, by taking the N terms (N is fixed) with the highest scores as research interest labels. We can also use the scores for calculating a similarity between the research interests of different people. This is used as a criterion to evaluate groups (cf. section 6.2): A group is valued higher if the pairwise similarity between all members is high.

Group labels are generated basically in the same way as individual research interest labels: All abstracts by all group members are concatenated and treated as a single document.

[3]OpenNLP maximum entropy POS tagger: `http://maxent.sourceforge.net`

For roles, we do not have the possibility of extracting labels from our publication corpus. As the roles are automatically generated instead of predefined, we also cannot manually assign labels without inducing bias. To at least provide a basic role vocabulary for ASR, we manually extracted all role labels from the official personal web sites. These were not assigned to any role.

7. Prestudy

7.1. Goals of the Prestudy

To find out how potential users would react to questions as planned for the dialog component of the IslEnquirer system, a user study was carried out. In total, twelve researchers, professors and students working at two different institutes were interviewed using a two-part questionnaire. The first part covered questions we intended to use directly for the robot system. The second part consisted of a list of people associated with the institute and participants were asked to assign a short role label to each name. This part was added to investigate the conception of roles people had in mind. Participants were told that the questions were meant to be used by a fully autonomous robot to interview them about their work. They were further informed that their contributed information will be kept anonymously.

The main goals of this investigation were:

1. To check the general acceptance of personal questions concerning work, roles and professional relationships
2. To find out how different users answer questions differently in terms of length, granularity and detail
3. To identify questions that are hard to answer or hard to come up with a short enough answer understandable by the system
4. To better understand the participants' conception of roles

7.2. Results

The main observation we made during the prestudy are:

Establish trust and understanding The feedback to the presented questions was mixed: most people were confident answering them, only one participant felt uneasy during the interview. On the other hand, we encountered people who enthusiastically shared their knowledge. Some participants were sceptical or unsure at first as they did not fully understand the purpose of the questions. The consequence we drew from this observation was to include clear and prominent information the robot gives to first time users. In addition, they are asked to confirm whether they agree with storing the gathered information in the database of the robot.

Context free grammars do not cover all free answers The prestudy suggested that not all responses to free answer questions can be covered by a context free grammar of reasonable size. This is caused by a large variety of wordings used to formulate the answers. Additionally, reflecting about the the answer caused many hesitations, corrections and false starts. Unparsable responses might though still contain information within the focus of the system. To acquire this information, we decided to make use of another speech recognizer, based on a n-gram language model. However, the potential users are all familiar with spoken dialog systems and their capabilities in terms of speech recognition and language understanding. It is therefore possible that their speaking style will adopt when confronted with a real robot instead of a human.

Explicit label questions do not work In its first design, the IslEnquirer contained dedicated label questions. Those were questions that explicitly asked the user to assign a generic term to a concept presented to him in form of a set of more specific terms. An example for this type of question is "What do you associate with the following keywords: corpus, translation, statistical?", which we expected to result in answers like "machine translation". However, this question did not produce viable results, for various reasons: for many participants, the keywords were to difficult or far-fetched to find a label for. For domain experts, the labels were too trivial or omnipresent. This observation led to the abolishment of label questions. Instead, the system now implicitly tries to gain information on suitable labels by observing their use in answers to other questions.

Expect answers of different granularity and semantic In the second part of the survey, participants were asked to assign roles to each entry of a list of people. Many of them are working at the institute, but some were former members, guests or not part of the scientific staff. The assigned roles were diverse, ranging from very differentiated to very coarse.

Some participants concentrated on the research interests to assign roles, some roles were based on personal relationships. The majority of participants respected the formal academic hierarchy, however on different levels of detail. Students had no insight in the role structure at the institute. To account for this diversity in answers, we stress the importance of giving good examples to the users during the dialog. We expect this to lead to a greater accordance among the users' role assignments. To process answers which assign a research group when the question asked for a role description, we introduce the possibility of ascribing more than one attribute to a question.

Avoid abrupt topic changes and repetitions Participants were often confused when the topic of the questionnaire changed rapidly, especially from user modeling questions to label questions (see above). This leads to the conclusion that it is important to make clear transitions between different question types to allow the user to focus on the new topic. A similar confusion effect was observed for questions that apparently repeated previous questions. The dialog system should avoid such repetitions.

8. Interactive Learning

This chapter describes the main parts of the interactive learning component of the IslEnquirer. It is a spoken dialog system which first identifies the user and then starts interviewing him to build and update its social user models. The purpose of interactive learning in this scenario is to verify the hypotheses created during the offline step (cf. chapter 6) and to extend and update the database by acquiring new labels and information that was not present in or extractable from offline data.

8.1. User Identification

The IslEnquirer depends on accurately determining the user's identity and name. New names must be learned for future interactions. For many people, the database already contains information which was generated during the offline step. We therefore cannot simply distinguish between known and unknown users but have to extend this concept by introducing a third user state. To the IslEnquirer, people can either be *known*, *unknown* or *unseen*. For known people a database entry exists and they were recognized by the system at least once. People who are unknown are not represented in the database and the system has no initial information on them. Students and new institute members are a common example of unknown users. Learning their identity and gaining information on them is hard, as we usually have to rely on spelling for names and open questions. Unseen users also have never communicated with the robot before, but are already contained in the database. They differ from known users as they are identified as "unknown" by the multimodal user id, but the system can use the existing information to facilitate name learning.

The strategy for name learning is as follows: We learn first name and family name separately. We assume a name to be learned if we received a certain number of confirmations (usually, this number is one). The name for which confirmation is requested can be obtained in multiple ways: If we are still learning the first name, the user identification component (capsulated in a user model[1]) is asked for the most likely user id which is mapped to a full name for confirmation. This hypothesis is based on facial and vocal cues [14] and comes with a confidence value that is checked against a threshold. If it is available and the confidence high enough, this is the most elegant way to get both first and family name at the same time. This may even happen in the very first turn, when the system greets the user with his presumable name. If the user id is too weak or unknown, we ask the user to first say and then, if this fails, spell his first name.

If we already learned the first name but still lack the family name, we resort to a different strategy to not overwrite the learned information: We generate all names in our database that fit to the first name ("fitting" means having an normalized editing distance of less than a threshold). This list is then weighted by the score assigned by the user model, given the models confidence is above a threshold and given that the best user model hypothesis is not labeled "unknown". Furthermore, the system prefers those hypotheses with a high interest value and which belong to a person that was not often seen yet. The reason behind the latter is that frequent users will already be trained well by the multimodal id component and will probably never come to this point.
This family name is then used for confirmation. On rejection, this procedure is repeated a few times (or until the list of possible names runs empty) before we assume to talk to an unknown user. If that is the case, we have to ask the user to say or spell his family name.

As we store only one name per person at the moment, we have to make sure that the name is always understood in the same way. This is accomplished by comparing the learned name to all stored names using the Levenshtein editing distance. If we find a match which is close enough (the distance does not exceed a threshold), we change the learned name to the matched name. Otherwise, we create a new person entry and enter the dialog module for new user registration. This module acquires basic information on the user, most importantly whether he is within the scope of the IslEnquirer scenario, i.e. an institute member or a student. If the new user is a student, the system asks him to name a contact person to whom a strong importance connection is established.

[1] note that the term user model is used for the identification component for consistency with other publications. The social user model that is acquired during the dialog is stored in the attribute containers described in chapter 4

8.2. Interactive Learning of Social User Models

In this section, we will describe all aspects of the main part of the IslEnquirer dialog manager which is designed to acquire information on social user models.

8.2.1. Question Types

Each question in the IslEnquirer system is assigned a question type. This type influences the usage of the question during dialog. To determine the type of a question, it is classified along the following two axis:

- **open vs. closed questions** open questions ask the user to formulate a free answer which is then parsed by the NLU to extract a label. Closed questions, also called confirmation questions, on the other hand propose a label hypothesis which the user can either confirm or reject. Open questions are intended to acquire new information in an unbiased way, while confirmation questions validate existing knowledge. At first glance, closed questions are more restrictive and less informative than open questions. They can however propose attribute values that the user would not come up with by himself.

- **direct vs. third person questions** while direct questions deal with the current user himself, third person questions ask the user on information about another institute member. The reason for the latter is to get additional sources of information and to acquire information on people who do not regularly visit or talk to the robot.

	open	**closed**
direct	Who is your most important coworker?	Are you researching on `[label]`?
third person	What is `[target]`'s role at the I S L institute?	Is `[target]` in the `[label]` research group?

Table 8.1.: Examples for questions of all four basic types

Additionally, there are questions for special situations. *Existence questions* and a *relevance question* are both closed questions. The former are used to verify the existence of a concept without implying its association to the user: "Is there a `[label]` research group at the ISL institute?". The latter is used to check the status of yet unseen people from the database: "Does `[target]` work at the ISL institute?". This is useful to remove nominal members or wrongly inserted person entries from the database.

8.2.2. Question Generation

Each question subdialog consists of up to three possible system utterances which contain a question: A basic question to begin the subdialog, a repetition question which can use a different wording if the basic question was not answered in an understandable way and a reconfirmation question to assert that the user's response was understood correctly.

Questions in the IslEnquirer are stored in a generic way. We only have to specify what texts are associated with it and of which type the question is (e.g. third person question or confirmation question). Some questions, especially those for confirmation, have variable content, which is set on runtime. This allows us to specialize one question with all different hypotheses we have for one attribute.

When asking for confirmation of an attribute with different labels, we have to select one of them to propose to the user. This is done in a randomized way to enable using different labels when the question is repeated (as opposed to always using the best label). This allows us to cover a larger variety of labels. The probability with which a label is selected is proportional to its normalized confidence score. For third person questions, we also have to set the target person of the question.

Finally, we attach the current target's attribute container to the question. Every question is associated with an attribute type for which it is valid. As not all attribute concepts are interpreted in the same way by different users, there can be additional associations. For example, our prestudy showed that some answers to questions on the role of a person actually dealt with its research groups. Associating multiple attributes allows correct processing of those answers (i.e., the response will be inserted in the container for groups, not roles).
Questions are conveniently stored in the database and extracted from there. This makes adding another question a very simple task, as the whole processing logic is independent of the actual question wording.

For an example of question generation, regard the following entry of the question list:

`GROUP_QUESTION_3` = Is `[target]` a member of the `[hypothesis]` research group?.

`[target]` and `[hypothesis]` are parameters that are set during question generation: The former is set to target person's name, the latter to the confirmation attribute.

The whole process of question generation is encapsulated in a *question factory*. The factory reads the questions from the database and creates a prototype of each one. During question selection, the factory returns the set of valid questions for a given setting by copying the eligible protypes and configuring their individual parameters. This provides a flexible mechanism to select an appropriate question set for every situation.

8.2.3. Question Selection

Every user can only spend limited time talking to the system. As we interrupt the user on his way to or from work, it is crucial to keep the dialog short. Therefore, we cannot pose every question that is available to the system but we have to choose the "best" questions, based on certain criteria. Foremost, we need to pose questions, which give us as much information as possible. Additionally, we want questions that are known for a very high success rate, which means that they are easily understood by the user and the responses are most probably covered by our grammar. Another important goal is not to bore the user by repeating the same questions and topics over and over again.

To cope with different, often adversarial goals during question selection, we use a flexible, modular scoring approach: For each question q and criterion c, a score $s_{q,c}$ in the interval $[0, 1]$ is calculated. These scores are multiplied and we select the question $\hat{q}$ with the highest overall score:

$$\hat{q} = \arg\max_{q \in Q} \prod_{c} {s_{q,c}}^{w_c} \tag{8.1}$$

To express the different importance of different goals, weighting is employed to scale the corresponding partial scores. The weights w_c are fixed for each factor.

We will now list the criteria which are implemented in the IslEnquirer system for question selection:

Expected information gain

The system needs to pose questions dealing with topics or persons it is still unsure about. Gaining information on these will (according to the update formula shown discussed in section 4.3) give the largest increase in overall confidence. This score is therefore based on the confidence of the attribute container associated with the current question. We distinguish two cases: For an open question, the score is simply negatively correlated to the confidence value, promoting questions for which a parsable user answer will have high impact. For confirmation questions, this approach will lead to selecting topics for which the system most probably can only provide low quality hypothesis for confirmation. This could lead to confusing questions, e.g. "Are you working on *features*?". Ill-formulated questions will cause low user acceptance and confuse the user in what the system is capable to understand.
Therefore, we support attribute containers of medium confidence for confirmation questions. The less the confidence deviates from the mean value, the higher we set the score. This results in a compromise of reasonable questions that still yield a significant confidence improvement. Before confidence scores are calculated, a normalization procedure is executed to ensure that this score is comparable to all other scores taken into consideration.

Question reliability

Our prestudy has shown that not all questions are equally likely to evoke useful, understandable answers. As this observation is user dependent and for some attributes we simply do not have enough high quality questions, we cannot simply discard all but the most reliable questions. Instead, we keep all questions, but assign a reliability score to each one. This score can for example be initialized using results from a Wizard-of-Oz experiment or by simply assigning default values for open and closed questions. The latter method is used for the IslEnquirer at the moment.
Every time a question is completed, its reliability gets updated. Aborting

a question leads to a reliability decrease, successful answer processing to an increase. For details, cf. section 4.3.1.

Topic and question frequency

As the system wants information from the user and depends on his cooperation, it is necessary not to bore him. Therefore, for every user, we keep track of the number of times each question and each attribute type (which can be covered by multiple questions) was mentioned during all dialogs. We include a score which is reciprocally proportional to this frequency.

Topic satisfaction

This score takes the same line as the previous one. Whenever the user successfully completed a question subdialog, its associated attribute type is flagged as satisfied for the rest of the dialog. Questions dealing with the same topic are than penalized with a fixed score < 1, as we expect to have already included the user's knowledge on this topic and further questions will only annoy the user or reduce his trust in the robot's capabilities. This behavior is in accordance with our observations during the prestudy: Users were confused when they were confronted with a question they regarded as equivalent to a previous question and were uncertain whether they should repeat their answer.

Topic continuity

Like the previous one, this is a score for indirect dialog flow control. When we abort a question (e.g. because we exceeded the maximum number of allowed turns), we penalize all questions dealing with a *different* topic with a fixed score < 1. By doing so, we expect to see another question on the same topic. This is motivated by our prestudy, showing that unexpected topic changes unsettled the participants and reduced answer quality. This is the case especially when the last question was aborted and the user seeks another way of informing the robot. Together with the previous two scores, this one allows us to influence the global course of dialog, without programming fixed rules like e.g. in a finite state machine.

Attribute relevance

Not all information is equally interesting. For example, we value information on research interest and relations among coworkers higher than participation in specific projects. This difference is expressed by another score to support questions dealing with important topics. This score depends on the attribute type associated to the current question. Relevance values were manually chosen during development. It is possible to automatically determine these scores by monitoring the usage of the generated web site and determining the frequency with which each topic is viewed.

Target importance

When using a third person question, the system needs to assure that the user is a reliable source of information on the question target. We therefore introduce the importance of the target for the user as another score. This will first promote direct questions as the importance score of the user for himself is defined to be one. Over time, when the social user model reaches a sufficient confidence, questions on the most important coworkers will emerge.

Target interest

We prefer questions on persons belonging to the "core" institute members before we deal with students or former members. This is expressed in the target's interest value. When interviewing people of low interest, this criterion will lead to more third person questions on their most important and more interesting contacts. Additionally, third person questions on people with low interest value are suppressed.

The robot should not ask third person questions on targets that are (not anymore) working at the institute as this could easily confuse the user with a name that is unknown to him. This leads to the introduction of a relevance question. It is employed when the best selected question is a third person question and the target user was never seen before by the system. The relevance question then replaces the original question and asks whether the target works at the institute. When rejected, the target's interest value and with it the score of the original question will drop. When confirmed, the original question will again have the highest score.

8.2.4. Question Dialog

The question dialog itself is contained in a separate module, which is reset for every new question. As the system has to handle uncertainty and give additional information to the user, a *question* is not defined as a simple question-response interaction but denotes a complete subdialog that consists of several parts of which some are optional:

- Introduction
- Example
- Basic Question
- Repeated Question
- Reconfirmation
- Completion/Abort

The following sections will cover these parts in greater detail.

8.2.4.1. Question Introduction

Before the actual question dialog begins, the system tries to warm up the user by smalltalk and question transitions.

Smalltalk is done before the first question in each session. The IslEnquirer contains a *smalltalk factory* that can generate and customize different smalltalk statements. Currently implemented are the following:

- Time related smalltalk: "It is a nice morning, isn't it?"
- A statement referring to the user who talked last to the system: "I just talked to `[target]`, he was very helpful!"
- A statement referring to the last attribute value the system successfully learned about: "Today, I already learned a lot about `[label]`"
- Smalltalk related to how often the user was already seen by the system: "You are helping me again, how nice!"

The architecture allows simple addition of new smalltalk statements, e.g. telling jokes or ranting about the weather. The customization introduces more variation and has the goal of strengthening the user's impression of an adaptive system and making the dialog more enjoyable. The latter is important as it is the robot that requires the user's cooperation, who in turn only has indirect advantages of providing information.

The second way of introducing questions is to make a question transition. These are statements like "Let's start with the first question" or "Just one more question". Their purpose is twofold: On the one hand, they clearly mark the transition between two questions and possibly between two different topics. The prestudy has shown that unprepared topic changes confused many users. On the other hand, these transitions make the dialog flow visible for the user, giving him feedback on the progress he made. This leads to a higher user motivation.

8.2.4.2. Giving Examples

The system makes ample use of examples as this is an unobtrusive way of communicating its capabilities and limitations. This takes place on two levels: On the first level, the user is implicitly informed what complexity the speech recognition and the language understanding are able to process (in our case, short phrases). He will also adopt the proposed formulations for open questions, where many responses are valid but only some of them are covered by grammar and vocabulary. These two factors will increase the recognition rate even for new users, as an example precedes the first question.
On the second level, the user learns more on the granularity and level of detail the system is interested in via examples. This is even more important than the first point: Most speech recognition errors are easily detected as the system will not be able to extract any of the expected feature structures. Additionally, if a misrecognition occurs, it is probably caught when the user is asked to confirm the understood information. If however the user gives information of wrong granularity (for example talking about technical details of his work instead of the general idea the system is interested in for displaying on the web site), only additional dialogs with other users or with the same user at a different time can compensate for this inconsistency. Giving examples at an early stage will diminish this effect and lead to a more consistent database. Examples are not only useful for database consistency, but will also give the user better

guidance for open questions and therefore a reduced cognitive load and a more comfortable dialog experience.

Examples are automatically generated from the set of all attribute values and their labels. All examples have the following form:

"The `[ATTRIBUTE_CLASS]` of `[OWNER_NAME]` is `[ATTRIBUTE_LABEL]`."

For example:

"The research interest of Hartwig Holzapfel is dialog."

The values are set according to three criteria, which are represented as values in the interval $[0, 1]$ and integrated via multiplication: Person interest value, attribute association score and label score. Preferred are questions where the target person is interesting (and probably well known within the institute) and the attribute has a high association score to the selected person, combined with a high label score.
To increase variability for repeated examples on the same topic, the system does not always select the best attribute value but does a randomized selection with a probability distribution derived from the example scores. The system will avoid examples that deal with the user itself or with the current question target. Those examples would either give no new information or bias the user.

After some dialog sessions, the user becomes accustomed to the system and examples become less important or even bothersome. For this reason, not every question is introduced by an example. Instead, there is a chance that an example is given. This chance is inversely proportional to the number of times the current user already interacted with the system. It is possible that the system cannot generate any example for a certain attribute type. This can happen when the offline initialization could not extract any labels and none were added by users, yet. In this case, the system notifies the user by saying that it "does not know much about this topic". We try to induce special care in the user's answer formulation by doing so.

8.2.4.3. Asking for Reconfirmation

The IslEnquirer has to cope with uncertainty. This is especially true for processing information gained from answers to open questions. As a grammar based speech recognition system can only give weak confidence scores, it is unclear if the extracted feature structure correctly represents the intended concept. Even if the ASR yields a perfectly accurate transcription (as it will usually be the case with "yes" or "no" answers to confirmation questions), there is still the danger of the user having misunderstood the question, either on an acoustic or semantic level. Asking for a confirmation is the usual solution [13], but it will lengthen the dialog, leaving less room for other questions. To draw a distinction to *confirmation questions*, which ask for confirmation of stored information, we call questions asking for confirmation of information from the current discourse *reconfirmation questions*.
The IslEnquirer uses the following decision rule for reconfirmation questions: They are applied if the concept extracted from the user's reply is not in accordance with the belief of the system. For confirmation questions, this is the case if the user rejects the hypothesis of the system, e.g. by saying "no". For open questions, this decision is made by determining the maximal score of the label in question over all associated attributes of the infered type. When this maximum exceeds a fixed threshold, no reconfirmation is requested.

A negative response to a reconfirmation question of a confirmation question will abort the current subdialog as the rejection itself already contains all information. For open questions, the discourse is cleared and the system uses the repetition question, which can have a different wording than the original question. It is possible that even after some repetitions and support via examples, a question cannot be answered. This may be the case if the user cannot come up with a formulation that is understandable by the ASR and NLU units or simply because he does not know the answer. The system will therefore abort any question that takes more than a fixed number of turns, apologize and carry on with the next question. The same behavior can be triggered by the user by saying that he does not know the answer.

The IslEnquirer dialog has the same structure most of the time and some parts of it are repetitive. As the system depends on users who regularly share their knowledge with the robot, we introduce the *phrase* *generator*. The phrase generator is used to diversify phrases that are often repeated in the course of a dialog. For every turn, the generator randomly selects one text block out of a set of possible alternatives for each phrase slot. Table 8.2 shows some examples. The generator serves two purposes: Firstly, it makes the dialog more varied and enjoyable for the user. Secondly, it allows the system to compensate for potential TTS errors as the user can ask for repetition of the last statement and get another formulation he can hopefully understand better.

`[reconfirm_phrase]`	you are in the ... research group.	`[correctness_phrase]`
↓		↓
I understood that		Is that correct?
You said that		Is that right?
You told me that		Did I get that right?

Table 8.2.: Exemplary results of the phrase generator

8.2.4.4. Updating the Database

When the question is completed or aborted, the target's social user model is updated. Other conditions may trigger an update as well. The process of updating is described in great detail in section 4.3. There are four occasions in which an update of one or more attributes is performed:

- **Update on completion** When a question is answered successfully as either an expected response was received or a response was confirmed by the user, an update on this label is performed.
- **Update on rejection** Whenever a label, either proposed on the initiative of the system or after a reconfirmation question, is rejected by the user, an update is performed on this label. This does not necessarily indicate the end of this question subdialog, open questions can continue with repeating the question.
- **Update on abort** When a question subdialog reaches the maximum number of allowed turns without a satisfying result, it is aborted. In case of a confirmation question, this is charged on the proposed label and an update is performed to reflect this.
- **Update based on n-gram recognition** cf. section 8.4.1

8.3. Learning of new Labels

As we employ open questions that allow the user to freely mention words which are unknown to the system, the IslEnquirer must be able to learn new words and names. Learning of person names can mostly be achieved with the same techniques as for user name learning as described in section 8.1. Coworker name learning is also started when a person uses only part of the full name as response to a question, e.g. "Stefan" or "Professor Waibel". This partial information is passed to the learning module which then checks whether the database contains any compatible full names. Only if this fails, normal name learning is started. The system also tries to classify a given first name as male or female, using statistics on the most popular names for both genders. This allows more natural formulations like "Is [his/her] name ...?"

Label learning requires more adjustments of the learning procedure. Firstly, the learning module infers the attribute type from the discourse information and the current question. This information can then be used to refer to the unknown concept as e.g. "the research interest" or "your role". Secondly, the user is told to repeat only the name of the concept. This helps if earlier recognition attempts failed because the label was embedded in a phrase which was not covered by the grammar. If we still fail, we assume that the term is unknown and the user is requested to spell it.
We do not want the user to spell terms that are too long and hence spelling them would be error-prone and exhausting. The number of tokens in the phoneme hypothesis of a detected OOV word is used as an approximation of the number of characters. If it exceeds a certain threshold, learning of this term is not triggered. Instead, the user is asked to use a simpler wording. In many cases, this has the side effect of leading to the use of a word from the vocabulary that does not require learning at all.

8.4. Speech Recognition

8.4.1. Language Models

Like most spoken language dialog systems the IslEnquirer uses a grammar based language model for speech recognition. However, grammar based language models have some drawbacks: They will always yield a result matching the grammar rules, even if the utterance is not actually covered by them. Only the limited number of utterances contained in the language model can be recognized, other sentences will result in unpredictable behavior. For small domains, this is usually not a problem. However, the prestudy suggests that people will use very different formulations when asked to talk freely about their work. They will also speak in an ungrammatical way, with many restarts, hesitations and ellipses. Giving implicit direction with carefully selected questions and employing examples will help to provoke parsable utterances but the coverage will never come close to 100%. As a grammar can only assign the probabilities 1.0 (if parsable) or 0.0 (not parsable) to a sentence, the probabilities produced by the system are not as differentiated as with language models that support a broader range of possible scores. This makes the resulting probability less useful for interpretation as a confidence score.

Another well known class of language models uses n-grams. There have been approaches to combine grammar and n-gram language models into one [29]. For the IslEnquirer, we take a slightly different approach as we do not need to have perfect recognition rates but need only to detect the keywords a user mentions in his utterance. When posing open questions during knowledge acquisition, there is one grammar and one n-gram recognizer running in parallel. The semantic of the user's response is still extracted from the grammar based recognizer. However, if the recognizer (erroneously) returns a feature structure different from the expected response, or if the recognized label was already rejected during this question or the recognized phrase contains an OOV word, the system does not automatically repeat the question. Instead, the n-gram hypothesis is searched for all labels known to the system. All matches are collected in a set. From this set, the system extracts one entry at a time and uses it for a reconfirmation question. The labels are extracted in the order induced by the label score for the current target's attribute container. This means that labels not assigned to the target will only be used if no other unrejected labels are available. When the user approves, a normal update is performed. Additionally, the system will do another independent update for all spotted keywords in the recognized utterance because the subdialog will only continue until the first match was confirmed by the user. Doing an update on all labels maximizes the information gain for the parsed phrase. The response processing procedure (cf. section 4.3.1) allows to give these updates a small impact, as the labels were not confirmed.

8.4.2. Vocabulary

The vocabulary of a speech recognizer determines all words it can possibly return as a result. The basic vocabulary consists of all words contained in the grammar, which includes the extracted labels from the offline step and all new labels learned during dialog. Additional vocabulary for two-step recognition (described in section 3.2) is formed by terms that were extracted from the publication corpus but not selected as labels.

The vocabulary of the n-gram based recognizer was chosen from the same texts that were used to train its language model: the first source is the concatenation of all publications used for the offline step. Instead of taking only the abstracts into account, here the whole documents were used, as the purpose of the n-gram recognizer is to extract a broader variety of formulations. The second source was the ISL web site, automatically downloaded and all meta information stripped using a HTML parser[2]. In addition to that, several human-human dialog corpora on technical issues[3] were used to extract a vocabulary.

To simplify the use of project and role names – for which no label hypotheses are created in the offline step – we manually extract all mentioned labels from the official ISL website and the directly linked personal home pages. Note that they were not assigned as labels for any person so the users still have the freedom to chose any label they find appropriate.

8.5. User Adaptation

The vision presented in section 1.2 describes a robot that can adapt its behavior to the social state of its users. Although most of these ideas are out of the scope of this work, the IslEnquirer system still adopts to its users to provide a better dialog experience. The following list summarizes the effects that our user models already have on the dialog:

- For experienced users, the dialog contains no lengthy introduction, shorter dialogs and less examples as they do not require much support during the interaction. This helps them to concentrate on the essential parts of the dialog and keeps it as short as possible.

[2] `http://htmlparser.sourceforge.net/`
[3] `http://www.icsi.berkeley.edu/Speech/mr/`

- For the purpose of question selection, the dialog strategy inspects the social user model. When using confirmation questions, it selects questions dealing with attributes that are presumably associated with the current user. This has two advantages: Firstly, the user will be more comfortable when asked about familiar topics and it will increase his trust in the learning capabilities of the system. Secondly, the system has a higher chance of acquiring high quality information.
- For third person questions, the strategy selects question targets that the user is familiar with, according to its importance container. The benefit is similar to the one for the previous point.

8.6. Session Replay

Every dialog session is stored in a special log file format. These log files contain all information required to reconstruct all updates triggered during the session. The IslEnquirer contains a session replay tool to reproduce the stored updates. Replaying all recorded sessions will result in the same database the online system produced[4]. The update representation in these log files is independent of the state of the database. This allows using recorded data to adjust or even replace the updating algorithms without the need to re-record sessions. This mechanism also makes it possible to update the offline database, for example, when the manually gathered publication corpus is replaced by one automatically created by the IFinder [17]. This will of course change the resulting database, but as long as the attribute types are unchanged and no questions are removed, the result will be no different from a live session containing the recorded answers.

Another important aspect is an increased flexibility in evaluating the database, e.g. to produce a time series of updates by taking a snapshot after each session. This can be used to investigate or display the development of attribute values over time. In contrast to doing this during the recordings, we can select a subset of sessions or simulate the effect of hypothetical updates if no appropriate or not enough real data is available.

[4]given that both the online system and the replay started with the same initial database

9. Experiments & Evaluation

9.1. Goals & Research Questions

We will now evaluate both main components of the IslEnquirer system. Before we begin with our analysis, we formulate the following research questions that should guide the evaluation:

1. Is it possible at all to gather social information using unsupervised network analysis and interactive dialog with limited speech recognition and understanding capabilities?
2. How well does the gathered information represent the real social identity of the participants?
3. How large is the improvement of the data gathered during dialog compared to the database with results from offline processing? What are the contributions of both components?
4. Was the information correctly understood, parsed and represented?
5. How is the system perceived by the user?

9.2. Challenges of Evaluating Social Information

A challenge in evaluating the social data we collect is that there is no objective metric to capture the quality of the data. Even if we present the data to the concerned people, we can only check whether the data is conform with their own self image. However, the latter does not necessarily represent the "real" circumstances but can be tampered by overestimation, modesty or other influences. Official publications (e.g. the web site of the institute) are also usually created by a small group of members and therefore often biased. Additionally, they do not change often and contain outdated information.

To cope with this problems, we use the following approach: Wherever possible, we try to apply objective quality measures. This is for example done for dialog length and task success rate during dialog. When evaluating the social data we gathered, we resort to a qualitative (instead of a quantitative) analysis, doing a critical review of the data, combined with comparisons with other sources, when available.

9.3. Evaluating Social User Models

During the offline step, we processed a total of 225 publications of which 177 came with a directly accessible full text. Of those, 142 were classified as English texts and used for keyword extraction. All used documents were found on the official ISL publications web site and on personal web sites linked in the *who is who* section of the ISL web site. Only gentle preprocessing was applied to the list of publications to achieve a uniform list entry format.

To evaluate the dialog component of the IslEnquirer, we recorded a total of 39 sessions with a total of 19 participants. The user group comprised of students, researchers and a group leader. All participants were associated with the ISL institute and therefore familiar with autonomous systems. No participant was a native English speaker.

To investigate the quality of the data collected during the dialog uncoupled from errors that occurred during dialog processing, we created a modified database based on the real data collected during live dialog sessions. All corrupted log file entries resulting from errors in dialog management are repaired or removed. Errors caused by recognition errors, misunderstanding of speech synthesis, misconceptions of users, unparsable utterances or wrong information intentionally provided by the user are not corrected. Every dialog system must cope with these types of errors.

9.3.1. Person Entries

The foundation of the IslEnquirer database is the set of persons for which a social user model exists. As we do not start with a list of active institute members, our approach has to extract the relevant entries by itself. Out of 177 unique names in the publication corpus, 27 were selected to form the initial database. This results in a recall of 1.0 and a precision of

0.44 when comparing to the set of institute members without students. A high recall is more important than a high precision, as a lack of the latter can easily be mitigated during the dialog using relevance questions, cf. section 8.2.1. In fact, during the recordings no user was asked questions about non-members: Either their own user model still required updates or their strongest importance ties connected them with other active members. On the other hand, leaving out an active member is more harmful as it requires the system to learn his name from scratch and skips the opportunity of pre-filling the database during the offline step.

We can compare these figures with two other sources. The ISL web site links personal web sites for their listed members. Only when a non-empty personal page is given, we count it as an existing person entry. This leads to a precision of 1.0 for the web site, however the recall of 0.6 is very low, as not every institute member maintains a personal web site. For the Arnetminer [42], an automatically generated online database of the scientific community, the recall even drops to 0.53. This may be caused by the wider scope of Arnetminer but reduces its utility for studying single institutes. The precision for Arnetminer is not defined as it tries to cover the whole scientific community.

During the course of the dialog, nine new person entries were created. 77.8% of them were correct, the rest corresponds to misrecognized names of known entries which were wrongly confirmed.

9.3.2. Groups

There exists no trusted or consistent source of information on the research groups within the institute. The dialog experiments and interviews during the prestudy (chapter 7.1) however suggest that the majority of users agrees on the existence of such research groups and that there are the following likely candidates: speech recognition, computer vision, machine translation, dialog. We will now compare our results, called *offline group set*, with these expectations.

Our offline component identifies eight research groups. The *computer vision* group is perfectly represented with a recall of 1.0 and a precision of 0.875. This takes into account all scientific members (not counting students) working in the same part of the building and all researching on or working with visual perception for interactive systems. There is

a *face recognition* subgroup (constituted by a separate web site[1]) within the vision group, which is also represented by the automatically extracted research groups. However, as it contains mostly students who did not publish many scientific papers and are therefore not present in the offline group set, the actual composition deviates.
A machine translation group is not represented in the offline group set. This is because at the ISL, only one researcher is dedicated purely to this field of study. There exists however a group in the offline set that can be labeled as *speech translation* group, working on the combination of ASR and translation components. There are at least two ongoing speech translation projects in which the institute participates[2], while most of the pure machine translation research is done at a partner university[3] where 38.5% of all researchers claim to be associated with machine translation in contrast to only 18% for the ISL institute.
As all speech recognition researchers are also involved in the speech translation projects, a separate speech recognition group is not present in the offline group set. This is caused by the merging or deletion of similar groups, which is based on the relative size of the intersection of both groups. We then tried another approach were two tentative groups were only merged when the result was of better quality (cf. section 6.2). This solved the problem of the non-existence of a dedicated speech recognition group. Additionally, most other groups described here remained unchanged or were even cleaned from non-core members. In return, the number of entries in the offline group set increased by 25%. This indicates a general tradeoff between *internal noise*, resulting in fuzzy group definitions and additional non-core members, and *external noise*, which leads to more, often redundant, groups.
Until now, we only studied the groups we expected to see. We can however also identify another entry in the offline group set which does not correspond to a group of specialists working on a common field of research. Instead, it represents researchers from all parts of the institute coming together for a joint long-term work on *humanoid robots*. While this could be interpreted as a project (as some of the participants did), its length of over ten years justifies a classification as an unique research group. This shows that research groups can carry more information than shared re-

[1] `http://isl.ira.uka.de/face_recognition`
[2] TC-Star, Open Domain Lecture Translation
[3] Carnegie Mellon University, Pittsburgh (CMU)

search interests.

By subjective assessment, the extracted group labels are of lower quality than the terms extracted for individuals. This is caused by two different problems: Firstly, the terms for research groups should be more generic than those of the individual members. However, this requires terms that are not used in the specialized papers published by the members which are our only source for labels. Secondly, there are research groups that are not formed by people all working on the same general research area, e.g. the humanoid robots group discovered above. These groups get labels that are a union of all specialists' labels. An improvement might be to weight the group documents with the degree of overlapping of its authors with the set of group members. This should increase the influence of publications with all members participating. These are usually general project descriptions an might contain suitable labels.

After incorporating data from recorded sessions, we repeated our analysis. The most important improvement is the addition of a designated dialog group which mainly consists of students working on the dialog component of a humanoid robot. Regular meetings indicate that this is indeed a distinct research group. The other groups remain structurally intact and are supplemented by adding new members. As we were already content with the group structure gathered during the offline step, this is reasonable behavior. The greatest impact the dialog sessions had on the existing groups was the modification of label assignments. For the newly created group, appropriate labels were gathered and for others more general labels were added. For at least one existing group, the best and most often assigned label ("vision") is not part of the research interest vocabulary and was not contained in the original group label list. This indicates that groups are indeed a generalization of research interest. No user mentioned the interdisciplinary group we identified above and due to the lack of appropriate labels, the system could not successfully propose this group during dialog.

9.3.3. Roles

For evaluation purposes, we present the official role distribution available on the ISL "who is who" page[4] in table 9.1.

[4]extracted at the end of january 2008

Role	Role Members
Director	Waibel, Alex
Associate Director	Schultz, Tanja
Faculty (CMU)	Vogel, Stephan
	Yang, Jie
Faculty (ISL)	Stiefelhagen, Rainer
Research Assistant[5]	Bernardin, Keni
	Danninger, Maria
	Ekenel, Hazim Kemal
	Fügen, Christian
	Holzapfel, Hartwig
	Kolss, Muntsin
	Kraft, Florian
	Nickel, Kai
	Stüker, Sebastian
	Voit, Michael
	Wölfel, Matthias

Table 9.1.: Official list of all members at the ISL institute and their roles

This table is not a perfect representation of the role structure of the institute, as it actually presents the members of a cooperation the institute is part of. Some information is outdated as at least three people on the web site changed their status or affiliation since the last update. This indicates that even official documents are no gold standard for the evaluation of social information. There also exists the role *researcher*, which denotes postdoc researchers. For the ISL institute, this role is empty and during all data collections, no participant differentiated between *researcher* and *research assistant*.

We also collected information on roles during the prestudy. We transform this collection to role assignments in the following way: For each questionnaire, all people that are both mentioned in the prestudy and are part of the IslEnquirer corpus are clustered in role classes based on similarity of the assigned labels. We list a cluster as a role if the majority of prestudy participants agrees on its composition (but not necessarily its label). The results are given in table 9.2.

We will now compare the automatically extracted results to the roles defined by external sources. As there exists no useful metric for compari-

Role Labels	# Members	Agreement
boss, principal investigator, professor	1	100%
group leader	1	100%
research assistant, PhD student, colleague	6	66%
role unclear[6]	1	66%

Table 9.2.: Roles extracted from the prestudy

son, we will again resort to a qualitative analysis.
We first investigate the results of the prestige algorithm (cf. chapter 3.8.5) which is used for role assignment. Note that the term "prestige" is just a name for a mathematical concept and cannot be used to draw conclusions to prestige in the common understanding. Table 9.3 shows that the ranking reflects the hierarchy extracted from the web site: Directors and faculty members form the top of the list, associates (i.e. members from other universities or research institutes) are found at the bottom of the prestige ranking. Research assistants make up the middle of the list. This shows that prestige is a good indicator of role affiliation based on hierarchy.

The complete role assignment algorithm creates five roles and assigns a unique role to seven persons. Of them, four also have a unique role assigned on the ISL web site. The remaining three are research assistants. Their unique role is due to their high prestige rank. Two of the non-trivial groups consist of research assistants with a precision of 1.0 if the members of the partner university are allowed, 0.79 if not. With 0.63, the recall is not very high. It has to be noted though, that not all wrong associations are equally severe. The three research assistants with a unique role can be integrated in the correct class by a single operation, which can be triggered by using a fitting label to describe the role during dialog. Extracting wrongly assigned members from a non-trivial group requires more operations. Thus, a high precision is desirable.
We call the other extracted roles *associates and former members*. This is not a very precise assignment and does not explain the existence of three groups instead of a single one. For our scenario and given that the collected data focuses on the ISL institute members, this is not a problem. The precision is 0.9 with only one ISL institute member being in the wrong role class.

When examining the data after the dialog sessions, table 9.4 shows a great consensus among the users. This was strongly influenced by giving

Prestige Rank	Role (from web site)
1	director
2	research assistant
3	faculty
4	research assistant
5	associate director
5	faculty
6	research assistant
7	research assistant
8	research assistant
9	research assistant
10	research assistant
11	research assistant
12	research assistant
13	research assistant
14	former member
15	research assistant
16	researcher
17	research assistant
18	faculty
19	research assistant
20	research assistant
21	research assistant
22	associate
23	research assistant
24	associate
25	associate
26	research assistant

Table 9.3.: Prestige ranking compared with predefined roles

examples of previously assigned role labels. Only one participant (an associate to the ISL institute) could not assign a meaningful label, one label ("group leader") could not be extracted as it was contained in a phrase not covered by the grammar. Two other labels were tried ("member", "mitarbeiter") but later replaced successfully.
Compared to the offline role classification, a whole new group called *students* was added (with perfect precision and recall). It contained only new entries as no students were included in the original database. For all roles of which at least one member participated in the dialog recordings, at least one label was extracted. There is no perfect agreement of the final roles with the official allocation, especially regarding the terms "researcher" and "research assistant". Judging from table 9.1 and the prestudy, both labels identify the same role class, which supports our decision to allow multiple labels for each attribute. We also see no use of the "faculty" label (which is uncommon in Germany), while the term "group leader" is not present in the official presentation but widely used by participants. This leads to the conclusion that no single source – even an official one – can cover the "real" role assignment and labeling.

Role Label	# Members
student	5
researcher	7
research assistant	6
phd student	1

Table 9.4.: Non-trivial roles identified by the IslEnquirer after the recorded dialog sessions

9.3.4. Reseach Interest

Research interest is one of the few exceptions where we can compare the IslEnquirer to related work. We will do so by juxtaposing the results of both the offline step and the dialog sessions with research labels we extracted manually from personal web sites. We also include the results of the Arnetminer web site [42]. Table 9.5 shows an overview of the obtained results.

Regarding the IslEnquirer, for all people there is at least one meaningful and general label among the top five (in most cases, top one) automatically extracted labels. Some lists contain garbage terms that are either in

person id	personal web page	arnetminer	IslEnquirer offline	IslEnquirer online
1	-	Speech-to-Speech Translation Continuous Speech Recognition Speech Recognition Statistical Machine Translation Connected Letter Recognition	speech recognition machine translation speech meeting dialog	speech recognition machine translation speech meeting dialog
2	Multimodal Human-Computer Interaction Vision-based Interfaces Visual Perception of Humans Visual scene and activity analysis Smart Rooms	Pointing Gesture Recognition Human-Robot Interaction head orientation 3-D People Tracking Model-Based Gaze Tracking System	head orientation face recognition pointing gestures tracking of attention	tracking head orientation face recognition pointing gestures of attention
3	-	-	color tracking combination body camera	tracking color combination body camera
4	Human-Computer-Interaction mobile phone interactions field experiments	-	availability communication people in and subject human factors	human factors availability communication people in and subject
5	face recognition systems	Multi-modal Person Recognition Multiresolution face recognition Video-Based Face Recognition Evaluation face recognition CHIL Project	face recognition face individuals features persons	face recognition face individuals features persons
6	automatic speech recognition machine translation simultaneous speech-to-speech translation	-	speech recognition speech translation meeting source language dialogue	speech recognition speech translation dialogue meeting source language
7	spoken dialogue systems cognitive architectures learning tasks multimodal systems	Dialogue Management Integrating Emotional Cues Interactive Space constraint-based multimodal fusion system	dialogue dialog intention speech grammar	dialogue dialog intention speech grammar
8	-	-	speech translation machine translation lectures speech recognition translation of	speech translation machine translation lectures speech recognition translation of
9	-	-	speech recognition subword units dialogue feature extraction participated in	speech language model speech recognition dialogue subword units
10	Visual person tracking Pointing gesture recognition	Pointing Gesture Recognition human-robot interaction Human-Robot Interaction Head Pose Head Pose Estimation	pointing gestures head orientation gesture recognition tracking stereo camera	pointing gestures head orientation tracking gesture recognition stereo camera
11	automatic speech recognition acoustic modeling multilingual speech recognition	-	grapheme speech recognition source language phoneme pronunciation	grapheme speech recognition source language phoneme language model
12	-	Head Pose Multi-View Head Pose Estimation Multi-view Approach Neural Networks Seminar Scenarios	head orientation recordings seminar frames estimating	head orientation recordings seminar frames head pose
13	Conversational Speech Recognition Robust Speech Recogntion MVDR Noise Suppression Far Distance Automatic Speech Recognition	ISL RT-06S Speech-to-Text System Interactive Space Microphone Array Driven Speech Warped-Twice MVDR Spectral Estimation Word Error Rate	mvdr speech recognition mit frequency das	mvdr speech recognition mit frequency das

Table 9.5.: Comparison of research interests assignments by different sources. Only the first five entries for every list are given

the wrong language (e.g. mit, das), some are of the wrong part of speech to form a useful label (e.g. participated in) or are to specific to a single publication or project (e.g. frequency, meeting).

At first glance, the results of online and offline result of the IslEnquirer do not differ much. However, the averaged confidence scores improved by a factor of 2.67 and the average difference between first and second label increased drastically from 0.03 to 0.31 score points. Our interpretation is that the labels which were generated offline already represented the research interest well. The dialog component was then able to confirm their quality and – where necessary – rearrange them. Good labels were confirmed and the entropy reduced accordingly.

The research interest labels after recording dialog sessions still contain some inappropriate labels. This is due to two different causes: Firstly, users generally used a single label to describe their research interest. This means that no additional labels will be acquired through user initiative. Secondly, caused by the improved confidence score of the first approved label and the fact that confirmation questions select the proposed labels based on scores, the system will not bring up any labels which are not on top of the hypotheses list. The benefit of the IslEnquirer architecture is that we can exclude labels with small scores to mitigate or completely remove this effect.

The system performs indirect updates on each attribute matching a used label. This is beneficial to update the research interest and other labels for users that do not often interact with the system and can therefore only contribute little amounts of data. This effect can be seen in the results for the labels "dialog" and "dialogue" which were used by many members of the respective research group. If one expects a major imbalance in representation of labels, a limit for this type of update should be introduced.

The personal web sites offer both general and detailed information on research interests, when available. As section 9.3.1 shows, this is only the case for 60% of all institute members. When no page is available, no information can be gathered at all. Another drawback is that research interest labels are usually contained in plain text and have to be extracted manually or with elaborated pattern classification approaches which would again introduce errors.

Data from Arnetminer is also not always available due to its broader focus. The data is usually more descriptive than the labels extracted by the IslEnquirer, but it also contains entries which are too specialized or of the wrong part of speech. Additionally, some entries are outdated.

To put these observations on a sound basis, we performed a study among all ISL members to determine the average approval of the extracted research interest labels. In total, we collected ten data sets. For those people who are represented in both the IslEnquirer and the Arnetminer database, we compare the outcome. The evaluation presented each participant a list of research interest labels associated to him. They were asked to rank each item in quality on a seven point scale from -3 to $+3$. In 75% of all cases which allowed a comparison, the average score for the IslEnquirer label list was better than or equal to the Arnetminer result. Averaged over all participants, the score for the IslEnquirer research labels is $+0.12$. The relatively low value is caused by inappropriate labels as we can see in table 9.5. The IslEnquirer however provides a ranking for its label assignment and our evaluation must account for this. We thus calculate a weighted average, where every score is assigned a weight proportional to its position in the person's ranking. This results in a much better score of $+1.05$. If we restrict the the evaluation to the first entry of each list, it further increases to $+1.8$. Additionally, including information from dialog sessions improves the weighted rating to $+1.22$ (a relative improvement of 16%) and the rating for best label only to $+2.4$ (relative improvement of 33%). The average for the Arnetminer labels is $+0.48$, which is largely due to one very good entry. If we exclude it, the average drops to -0.2.

90% of all participants have at least one label with maximal rating ($+3$) among the first three entries of the respective list. Several persons also ranked multiple entries with high scores. This indicates that the system should bring up known but unverified labels from time to time. This would allow to cover the complete list of labels, further increasing the improvement of the dialog part. Additionally, it would make the dialog more diversified for the user.

9.3.5. Other Attribute Types

For evaluation of importance connections, we find that 84.6% off all ISL members published their latest registered paper together with the coworker which is ranked first in the respective importance container after dialog. This shows that our database correctly reflects academic cooperation.

The project attribute type was completely ignored during the offline step as these are very hard to identify and extract from publications. During the dialog, four new projects were created with one to three members. In total, six project labels were assigned. This shows that attribute assignment without initialization is possible but requires more sessions as none of the stored projects contains all members. The named projects do not correspond to the project list on the official web site at all, as only one entry can be found in both lists. This indicates that projects underlie a fuzzy definition and that official information is not reliable to represent the actual understanding of the participants. We take this as another confirmation of our strategy to directly interact with people for building social user models.

The recordings for the mood attribute did not show the expected results. Apart from two sessions, all participants during the dialog recordings used a positive description of their feelings concerning their work. We assume that these questions were too intimate to answer faithfully, especially during an evaluation session that was recorded and watched by several bystanders.

9.3.6. Conclusion

The conclusion we draw from the observations of the user models from both online and offline step is that both components have their strengths and weaknesses and have to be combined for full effect. The offline component is strong in finding underlying structures for groups, roles and importance connections. Evaluating and changing these using interactive learning proved difficult, given speech-only interaction and limited understanding capabilities. However, the dialog shines when it comes to finding good labels to name the extracted structures during dialog and on the web site. While the offline processing resulted only in a rough and noisy label set, the dialog contributed clear labels for nearly every concept that came up during the recorded sessions. Dialog is also required to add new attribute values that were not contained in the offline corpus.

We still need the influence of both components on the whole database: The labels extracted during the offline step are required to form a vocabulary for ASR and a set of hypotheses for confirmation questions. The modification of attributes by interactive learning is required to add new and unknown members to their associated groups, roles and projects.

9.4. Evaluating the Dialog

9.4.1. Evaluating Objective Dialog Metrics

Table 9.6 summarizes the relative frequencies with which questions on certain attribute types were selected during the sessions. Additionally, we see how often an open question was chosen for a given attribute type. The frequencies largely correspond to the attribute relevance values which were defined manually. Importance is an exception, as the initial confidence scores were already high for the participating users. This also explains the small fraction of open questions for this attribute type as those are preferred if attribute confidence scores are low.
Third person questions were rarely used (2.8% of all questions) by the system as most users spent only one or two sessions talking to the system. With six different attribute containers, each maintaining multiple hypotheses, and a maximum of five questions per dialog session, the robot spends the first sessions with a previously unseen user by building the respective social user model. Thus, the evaluation of multi-source models has to be postponed to a long-term test.

Attribute	Relative Frequency	Fraction of Open Questions
Group	20.8%	52.5%
Occupation	18.8%	47.2%
Role	18.2%	57.1%
Importance	17.7%	17.6%
Mood	15.6%	63.3%
Project	8.9%	100%

Table 9.6.: Summary of relative frequencies of different question types

To measure the quality of the dialog system, we use the task success rate, where each question subdialog is counted as a single task. We define a subdialog as successfully completed when the user confirms the learned attribute or when the system correctly assigns the right attribute without (re)confirmation. For 185 question subdialogs, the task success rate is 74.6% by this definition. When we only regard confirmation questions, the success rate increases to 90.6%. This proves that confirmation questions are easy to answer and process and supports our effort to initialize the database before starting dialog sessions.
Out of a total of 105 open question subdialogs, 60% were successful. From the remaining subdialogs, we investigate the reasons for the failure: In 50%

of all cases using an out-of-vocabulary label – while the rest of the utterance was covered by the grammar – caused the failure of the subdialog. This should have been repaired by the ability to learn new labels, but we will see later that label learning was rarely successful. 30.8% of the failed subdialogs actually suffered from errors of ASR and only 17.9% of the failed dialogs, i.e. 6.9% of all open question subdialogs, did so because the user's utterances were not covered by the grammar. This surprisingly low value indicates that the users are able to adopt to the recognition and understanding capabilities of the system, especially when we compare this result to the findings of our prestudy in section 7.1. Note however, that the participants are computer scientists who have a better judgement on the skills of the robot than the average user. Still, most failed dialogs were caused by misrecognitions of the ASR component. This indicates that more work is required especially to make label learning more robust.

The average turn count for completed questions is 1.9, i.e. questions that get answered successfully come to this point after less than two turns on average. This suggests avoiding frequent asking for repetition. In the case of a repeated question, the user often assumed that he was not understood because of a speech recognition error, while his utterance actually was not covered by the grammar or contained an OOV term. This leads to several repetitions of the same phrase by the user without a chance for the system to understand. The consequence of this observation is that the system has to explicitly ask for changes in the user's phrasing to make clear what circumstance probably caused the problem. A better trained n-gram recognizer could also help in this situation by comparing both result strings to decide whether an acoustic misunderstanding or a coverage problem caused the error.

9.4.2. Evaluating Subjective Dialog Experience

The IslEnquirer system depends on users that are willing to share their knowledge with the robot. This makes it necessary to evaluate the subjective quality of the dialog system. Following the guidelines from [27], this is done by handing out an evaluation questionnaire after each interaction. This paper gives a catalogue of evaluation statements to asses the users' quality judgement. Each statement is presented with a seven point Likert scale [25] from strong accordance to strong rejection and is linked to certain quality components: acceptability, transparency, efficiency, cognitive

demand, cooperativity and task success. From the presented questions, 27 were selected and presented to the users directly after their interaction with the robot. Cf. A.2 for the full questionnaire.
The questionnaire contained many duplicate statements to avoid inserting a bias towards supporting answers. This made it lengthy and was criticized by some participants. Future studies might use questions instead, trading comparability[7] for shorter and more accepted questionnaires.
For presentation purposes, the scales of all statements have been normalized so that positive values always indicate positive feedback.

We have to take into account that all participants were computer scientists, which makes this study not generalizable to other user groups. We expect the participants to be experienced and confident in the interaction with autonomous systems, which we expect to result in a success rate above average. On the other hand, they come with higher expectations, are harder to impress by an autonomous robot itself and have a high sensibility for flaws.

During our recording sessions, we collected a total of 18 completed questionnaires. Two participants filled two quesionnaires on different days, the other users contributed one. The results are very diverse, from disappointed to enthusiastic. This is reflected by the high average standard deviation of 1.67 on a seven point scale. Another indicator is the used range for the user's overall impression: All values from -2 to $+3$ were used, with an average of $+0.69$. If we remove two sessions which were seriously hampered by technical errors in the early testing stage, the average overall impression increases to $+1.0$. We will now further investigate the most significant results of the study and try to identify reasons which caused positive or negative overall judgement.

On the positive side, we find that most users rate the system to be very friendly ($+1.73$) and the dialog to be short ($+1.71$), enjoyable ($+1.2$) and not boring at all ($+1.33$). Those are very important results as they promise long term acceptance of a dialog system acquiring social user models through interactive learning. They indicate that our efforts to keep the system entertaining for the users were successful and pay off by

[7]though the internal comparability can be maintained by using the same questionnaire over several projects

having willing users.

The users found it easy to keep track of the dialog flow (+1.75). On the one hand, this is a desirable result, which was aimed for by introducing and completing questions and by letting the robot always strive for dialog initiative. On the other hand, it results in a system which the user clearly does not have the control over (−1.19) and which is perceived as somewhat inflexible (−0.15). As most participants were first-time users of the IslEnquirer system, maintaining system initiative reduced the cognitive load and allowed them to concentrate on the questions and their answers. It remains to be examined whether experienced users will find a system with shorter dialogs and more possibilities of interaction more convenient. The high variance on whether the user felt strained (standard deviation of 2.05 with an average of +0.88) or relaxed (1.95, +0.07) indicate that the experienced cognitive load is strongly user dependent.
An important addition for achieving higher flexibility would be to allow more error correction strategies which the user to initiate corrections [13]. This would also improve the impression that mistakes are not always easy to fix (−0.5).

Negative feedback was usually caused by misrecognitions and misunderstandings: users felt not always well understood (−0.6) and the system was not perceived to make few mistakes (−0.65). As a result to recognition and dialog flow errors, the system did not always react as expected (−0.65). We identify three main causes for this judgement: Firstly, during the first recordings, the system suffered from a dialog behavior that caused the system to interrupt the user at unexpected occasions or unexpectedly aborted a question subdialog even when the system just asked to confirm the correctly understood statement. Results improved when this behavior was modified.
Secondly, as section 9.4.3 shows, recognition rates for free answers were low and the system was not able to learn many new words that were not part of the initial vocabulary. Improving recognition rates is hard to do in the dialog layer. Section 9.4.3 suggests to invest more in the combination of grammar and n-gram based ASR systems.
The third cause, which was often expressed during user interviews, was the low quality of the synthesized voice, especially for names for which pronounciation was generated automatically. Mistakes in this area lead to user confusion or unsuitable responses.

9.4.3. Evaluating the Recognition

9.4.3.1. User Identification

The user identification component showed satisfying results for our task. The average turn count is 4.49, while the lowest possible turn count[8] for first time users is 3. During the recordings, the system learned six new names, which usually had to be done by spelling first and last name. Without these sessions, the average turn count goes down to 3.1, as people in the database are proposed a family name and user for which models exist in the multimodal identification component are usually recognized in the first turn.
In 69.2% of all sessions, the user model confidence exceeded the threshold set by the dialog strategy, which then proposed the corresponding name to the user. The prediction was correct for 62.9% of those attempts. Errors were usually caused by unknown users who were wrongly matched a by a model trained with much data. In the course of time, the recognition rate decreased as more recordings were made and more models added. The nevertheless promising results indicate that tuning of the cooperation between dialog strategy and user model will further improve the identification dialog.

9.4.3.2. Speech Recognition

To evaluate the speech recognition quality, we calculate the *concept error rate* (CER), i.e. the relative number of utterances for which no or a wrong semantic concept was extracted. In total, we evaluated 169 utterances that answered open questions. With 60.9%, the CER for the grammar based recognizer is very high. When we group all identical responses for the same question together, it improves to 49.6%. Regarding the n-gram recognizer, the CER is 49.7% and improves to 41.3% when all repetitions of the same label are summarized. It has to be noted that this comparison is not entirely fair, as the n-gram based recognizer is not capable of extracting semantic concepts and it is not possible to pass its hypothesis through the Tapas grammar as functional words are often recognized wrong. The consistent use of our current extraction method – pattern matching of known concepts – would introduce more understanding errors. The high CER is a main cause of failed question subdialogs.
The task of the n-gram recognizer in our setup was to back up the nor-

[8] asking for first name, confirming first name, proposing suited family name

mal grammar based recognition. This was a successful approach, as we find that for 39.8% of all misrecognized utterances the n-gram recognizer correctly extracted at least one of the mentioned labels. This observation shows that both recognizers complement each other.

For some acoustic inputs, the grammar only covers part of the used label and slightly changes or simplifies the answer. For example, "language modeling" and "dialog management" are not covered by the grammar and are transformed to "language model" and "dialog". However, in many cases in which this phenomenon occurred, the user nevertheless confirmed the proposal of the system. The n-gram recognizer often extracted the full label. This behavior of confirming transformed labels simplifies the process of finding unambiguous names for attributes as it presents a natural way of mapping a user's variation of an existing label.

Spelling recognition for learning of OOV words resulted in a recognition rate of 16.6%. This low value is explained by the language model of the spelling recognizer – the only available one at the time of development – which was trained on German personal names and employed to recognize spelling of English concept labels. This clearly indicates that the learning component has to be improved. Even a perfect spelling recognition will not solve the problem of label learning, as we avoid asking for spelling of long words. Chapter 10 makes some suggestions for better label learning. Note that this failure does not mean that the assignment of new concept labels to attributes does not work. During the offline step, we created a vocabulary of 663 terms and associated 151 distinct terms (22.8%) as initial labels to attributes. All those labels can be used without requiring the user to spell.

10. Conclusion & Future Work

10.1. Summary

In this work, we present the IslEnquirer, a first attempt to provide a humanoid robot with social awareness. This is achieved by building social user models, which store information on the relations between people, the groups they belong to and their roles within the community. We describe how social user models are represented to be robust against noise and able to handle multiple hypotheses.

We then show how social user models are trained using a two-step approach: Firstly, we initialize the database with hypotheses which are generated by evaluating a publication corpus. Using methods of social network analysis and information retrieval, we automatically extract an initial social user model. This model is then verified and extended by human-robot dialog. A general dialog framework and our task specific strategy are presented. This dialog strategy is designed to choose questions that promise a high information gain and a high success rate.

By evaluating the collected data, we validate our hypothesis that social user modeling is possible using this combined approach. A user study shows that the dialog system is accepted by its users. Both results combined show that our social user modeling approach is successful and can be used to extend the understanding of human behavior.

10.2. Future Work

One main limitation of the current system is the very basic implementation of learning new concepts during dialog. Currently, there are two ways of inserting a new concept: The speech recognizer can recognize words that are not stored in the IslEnquirer database but have been seen during the processing of publication abstracts. However, this mechanism cannot cover newly emerging fields of study. The other way of learning is through spelling the label of the concept. A first improvement would be to integrate a spelling recognizer tailored for our task. Even then, not all labels are suited for being spelled: The current recognizer cannot recognize more than one word at a time and spelling long or unusual words will become tedious for the user. Another potential for improvement on the speech recognition side lies in a better combination of grammar and n-gram based recognizers. One application would be to compare results of both recognizers to distinguish between acoustic mismatch and lack of grammar coverage for misunderstood utterances.

Techniques from object learning could introduce a non-flat ontology to the concepts the system can understand. This would solve the problem of people using levels of granularity the system cannot cope with. Instead, the system could generalize from some specific labels a user comes up with or make an elaborated guess for the user's specialization based on a general label he employed. This would also make it possible to represent differences between people working on similar subjects as their personal research interest could be stored without losing information on their general field of study. Having a detailed ontology would also improve the speech recognition as many more labels could be inserted in the dictionary without an increased chance of confusion. Instead of allowing all possible labels as a response, the system sets its expectation to contain only the labels in a subtree of the ontology which is associated with the user. Additionally, a learning algorithm that separates descriptive modifiers from objects would make learning of variations much easier. For example, if the term "real-time tracking" is already known, the task to learn the new label "audiovisual tracking" no longer requires the system to learn the whole phrase but only the adverb "audiovisual".

The IslEnquirer would profit from better name learning and allowing multiple weighted hypotheses of names for person entries and importance connections. This would make the system more robust against name recognition errors. In addition, combined learning of first and last names could shorten the dialog and increase the user acceptance. First work on this is done in [36] and is ready to be integrated in the IslEnquirer framework. Distance identification without forcing the user to stand in front of the system for a moment would allow the system to decide whether a certain person should be approached by the system. This decision can be based on the information it is likely to learn or on the person's role (e.g., do not disturb professors too often).

Another step to take would be including a fully automatic version of the IFinder to end up with a complete chain of autonomous tools for information acquisition from searching the web to presenting and updating the information via dialog and web site.

To fully understand the acceptance, capabilities and limitations of the IslEnquirer system, more data has to be recorded and evaluated. Especially a long term observation of the system is necessary to monitor the processing of slow and sudden changes in the domain (e.g., a master student's promotion to PhD student status). To that end, it is advisable to integrate the social questions in a more general setup. Instead of going through a list of questions, only one or two questions should be presented in each session, among other content, to keep the user interested over a long period of time. The framework presented in this work makes it easy to perform those changes. The components for question selection and social user modeling can easily be reused independently of the current implementation.

The long term goal of the IslEnquirer should of course be to integrate and actually use the gathered information directly in the dialog as described in section 1.2. A possible approach would be to integrate a publication recommendation system or an expert finder. Offering services to the user would also further increase the acceptance of the system.

A. Appendix

A.1. Example Dialog

Robot: I am Robbi, the new I S L webmaster.
Do you have time to update my database?
User: Sure!
Robot: You are Hartwig Holzapfel, right?
User: That's right.
Robot: I will now ask you some questions about your work and your professional relations here. This will update the new I S L webpage. Are you ok with that?
User: Okay.
Robot: I just talked to Maria. She was very helpful!
Robot: Ok, let's start with the first question!
Robot: The research interest of Hazim Kemal Ekenel is face recognition.
Robot: What is your general research interest?
User: I am working on dialog management.
Robot: You told me that you are researching on dialog. Is that right?
User: That's correct.
Robot: Now it's clear to me. Thank you!
Robot: Is there a speech research group at the I S L institute?
User: Yes, there is.
Robot: I understand, thank you!
Robot: Let's move on to the next question.
Robot: What is Felix Putze's role at the ISL institute?
User: I don't know him.
Robot: I am very sorry. Lets try something different.
Robot: What is your role at the ISL institute?
User: I am a research assistant.
Robot: I understand. Thank you!
Robot: Let's move on to the next question.
Robot: Is Stephan Ultes an important coworker of yours?
User: Yes, he is.

Robot: I understand, thank you!
Robot: Just one more question!
Robot: What project are you currently involved in?
User: In the Humanoid Robots project
Robot: Sorry, I didn't get you. What project are you currently involved in?
User: Humanoid Robots
Robot: I understood that you are involved in the Humanoid Robots project. Did I get that right?
User: Yes
Robot: Good bye Hartwig. Thank you for your help!

A.2. Questionnaires

This section presents the questionnaires that were used during the prestudy and for evaluation. Some material is only available in German. For a translation of the evaluation questions, refer to the original source [27].

Questionnaire on possible interactions (free answers)

The following questions will be used by a robot in interaction with a human user to learn more about people at the ISL institute. Please give a short and spontaneos answer to each one (or mark the question if this is not possible).

Thank you!

What is your name? ______________________________

What is your academic degree? ______________________________

What is your role at Karlsruhe university? ______________________________

What is your role at ISL? ______________________________

What is your research interest? ______________________________

Who is your most important co-worker? ______________________________

With whom did you discuss your work lastly? ______________________________

Who is your boss? ______________________________

What research group are you in? ______________________________

What is the general subject of your current work? ______________________________

What is a good label for the keywords {corpus, translation, statistical}? ______________________

What project are you currently involved in? ______________________________

Evaluationsbogen zum Flurroboter

Bitte geben Sie für jede der folgenden Aussagen an, wie sehr Sie ihr zustimmen:

Gesamteindruck der Interaktion

sehr schlecht								sehr gut

Ich wusste zu jeder Zeit, was ich dem System sagen konnte.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Das System ist freundlich.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Das System reagiert nicht immer wie erwartet.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Ich würde das System in Zukunft wieder benutzen.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Das System machte wenige Fehler.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Ich fühlte mich angespannt während der Benutzung des Systems.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Das System reagierte zu langsam.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Ich konnte auftretende Fehler leicht beheben.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Es machte Spa ˚ , mit dem System zu arbeiten.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Es passiert leicht, dass man den Faden verliert, wenn man mit dem System spricht.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Die Interaktion mit dem System nervt.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Es war angenehm, das System zu benutzen.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Es ist klar, wie man mit dem System zu sprechen hat.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Ich hatte das Gefühl, dass ich die Kontrolle über das System hatte, während ich es benutzte.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Die Interaktion mit dem System ist langweilig.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Das System ist leicht zu benutzen.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Ich musste mich sehr auf die Interaktion mit dem System konzentrieren.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Die Interaktion mit dem System ist monoton.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Ich fühlte mich entspannt, während ich das System benutzte.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Die Interaktion mit dem System ist frustrierend.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Das System ist zu unflexibel.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Das System reagierte wie ein Mensch.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Ich fühle mich gut vom System verstanden.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Das Gespräch war zu lang.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Ich bin insgesamt mit dem System zufrieden.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Die Benutzung des Systems hat sich gelohnt.

stimme ganz und gar nicht zu								stimme ganz entschieden zu

Question number	average score	standard deviation
1	0.69	1.21
2	0	1.6
3	1.73	1
4	0.65	1.61
5	0.25	2.41
6	-0.65	1.68
7	-0.88	2.05
8	-0.94	1.56
9	-0.5	1.73
10	1.2	1.56
11	-1.75	0.97
12	-0.71	1.81
13	0	1.75
14	0.06	1.68
15	-1.19	1.74
16	-1.33	1.81
17	0.35	1.49
18	0.63	1.76
19	-0.07	1.65
20	0.07	1.95
21	-0.47	1.75
22	0.15	1.83
23	-1.29	1.6
24	-0.6	1.58
25	-1.71	1.44
26	0.09	1.88
27	0.46	1.95

Table A.1.: Results of the subjective dialog evaluation. Numbers refer to the order of questions in questionnaire

Bibliography

[1] B. Balasundaram, S. Butenko, I. V.Hicks, and S. Sachdeva. Clique relaxations in social network analysis: The maximum k-plex problem. 2006.

[2] Albert-Laszlo Barabasi, Hawoong Jeong, Zoltan Neda, Erzsebet Ravasz, Andras Schubert, and Tamas Vicsek. Evolution of the social network of scientific collaborations. *Physica A: Statistical Mechanics and its Applications, Volume 311, Issues 3-4*, 2002.

[3] Ulrik Brandes and Jürgen Lerner. Role-equivalent actors in networks: Group structures beyond dense communities. *Proc. Workshop Social Network Analysis and Conceptual Structures (Intl. Conf. Formal Concept Analysis, ICFCA'07)*, 2007.

[4] Catherina Burghart, Ralf Mikut, Rainer Stiefelhagen, Tamim Asfour, Hartwig Holzapfel, Peter Steinhaus, and Ruediger Dillmann. A cognitive architecture for a humanoid robot: A first approach. *5th IEEE-RAS International Conference on Humanoid Robots*, 2006.

[5] Bob Carpenter. *The Logic of Typed Feature Structures*. 1992.

[6] N. Dahlback, A. Jonsson, and L. Ahrenberg. *Wizard of Oz-studies – why and how*. 1993.

[7] Matthias Denecke. Object-oriented techniques in grammar and ontology specification. *The Workshop on Multilingual Speech Communication. Kyoto, Japan*, 2000.

[8] Jill L. Drury, Jean Scholtz, and Holly A. Yanco. Awareness in human-robot interactions. *IEEE International Conference on Systems, Man and Cybernetics, 2003*, 2003.

[9] Hazim Kemal Ekenel and Rainer Stiefelhagen. Local appearance based face recognition using discrete cosine transform. *13th European Signal Processing Conference (EUSIPCO 2005), Antalya, Turkey*, 2005.

[10] Christian Fügen, Hartwig Holzapfel, and Alex Waibel. Tight coupling of speech recognition and dialog management & dialog-context dependent grammar weighting for speech recognition. *Proceedings of the International Conference on Spoken Language Processing, (ICSLP), Jeju-Island, Korea*, 2004.

[11] Gerhard Fischer. User modeling in human-computer interaction. *User Modeling and User-Adapted Interaction, Volume 11, Numbers 1-2*, 2000.

[12] Eugene Garfield. Citation indexing for studying science. *Nature, Volume 33*, 1970.

[13] Petra Gieselmann. Comparing error-handling strategies in human-human and human-robot dialogues. *Proceedings of the 8th Conference on Natural Language Processing KONVENS, Konstanz*, 2006.

[14] Philipp Grosse. *tba (Studienarbeit)*. 2008.

[15] C. Gutwin, S. Greenberg, and M. Roseman. Workspace awareness in real-time distributed groupware: Framework, widgets, and evaluation. *Proceedings of HCI on People and Computers XI*, 1996.

[16] Robert A. Hanneman and Mark Riddle. *Introduction to social network methods*. 2005.

[17] Ronny Händel. *PAT-Trees zur Automatischen Informationsextraktion aus semi-strukturierten Webdaten*. 2008.

[18] Hartwig Holzapfel. A dialogue manager for multimodal human-robot interaction and learning of a humanoid robot. *to appear*, 2008.

[19] Hartwig Holzapfel, Thomas Schaaf, Hazim Kemal Ekenel, Christoph Schaa, and Alex Waibel. A robot learns to know people - first contacts of a robot. *KI 2006: Advances in Artificial Intelligence, Lecture Notes in Computer Science, volume 4314*, 2006.

[20] Hartwig Holzapfel and Alex Waibel. Behavior models for learning and receptionist dialogs. *Interspeech 2007, Antwerp, Belgium*, 2006.

[21] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. 2001.

[22] Qin Jin, Tanja Schultz, and Alex Waibel. Phonetic speaker identification. *7th International Conference on Spoken Language Processing, Denver, Colorado, USA*, 2002.

[23] Robert Kass and Tim Finin. Modeling the user in natural language systems. *Computational Linguistics, Volume 14, Issue 3*, 1988.

[24] Filip Krsmanovic, Curtis Spencer, Daniel Jurafsky, and Andrew Y. Ng. Have we met? mdp based speaker id for robot dialogue. *Interspeech 2006 - Ninth International Conference on Spoken Language Processing*, 2006.

[25] Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology, vol. 140*, 1932.

[26] Bradley Malin. Unsupervised name disambiguation via social network similarity. *Proceedings of the Workshop on Link Analysis, Counterterrorism, and Security, Newport Beach, CA.*, 2005.

[27] Sebastian Möller, Paula Smeele, Heleen Boland, and Jan Krebber. Evaluating spoken dialogue systems according to de-facto standards: A case study. *Computer Speech & Language, Volume 21, Issue 1*, 2007.

[28] Friedrich Pfeiffer and Hirochika Inoue. Honda humanoid robots development. *Philosophical Transactions of The Royal Society, Volume 365, Number 1850*, 2007.

[29] Anton Prevosti. Semantic-based hidden markov/context free grammar language modeling, 2006.

[30] Thomas Prommer, Hartwig Holzapfel, and Alex Waibel. Rapid simulation-driven reinforcement learning of multimodal dialog strategies in human-robot interaction. *Interspeech 2006 - ICSLP, Pittsburgh PA*, 2006.

[31] Ramon Sanguesa, Alberto Vazquez-Huerga, and Javier Vazquez-Salceda. Mixed collaborative and cognitive filtering in multiagent systems. *3rd Workshop on Agent-Based Recommender Systems (WARS 2000), Barcelona*, 2000.

[32] Kristie Saymor, Andrew McCallum, and Ronald Rosenfeld. Learning hidden markov model structure for information extraction. *AAAI 99 Workshop on Machine Learning for Information Extraction*, 1999.

[33] J. Scholtz. Human-robot interactions: creating synergistic cyber forces. *Hawaii International Conference on System Science, Jan, 2003*, 2002.

[34] Hagen Soltau, Florian Metze, Christian Fügen, and Alex Waibel. A one-pass decoder based on polymorphic linguistic context assignment. *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2001.

[35] H Tajfel and JC Turner. *The social identity theory of intergroup behavior*. 1986.

[36] Stefan Ultes. *Lernen von Vor- und Nachnamen im natürlichsprachigen Mensch-Roboter-Dialog*. 2008.

[37] W. Wahlster and A. Kobsa. Dialogue-based user models. *Proceedings of the IEEE, Volume: 74, Issue: 7*, 1986.

[38] Stanley Wasserman and Katherine Faust. *Social Network Analysis*. 1997.

[39] Marek Wester. *Rapid development of object oriented semantic grammars*. 2005.

[40] Douglas R. White and Karl P. Reitz. Graph and semigroup homomorphisms on networks of relations. *Social Networks, vol. 5 (1983)*, 1983.

[41] Dekai Wu. Active acquisition of user models: Implications for decision-theoretic dialog planning and plan recognition. *User Modeling and User-Adapted Interaction, Volume 1, Number 2*, 1991.

[42] Limin Yao, Jie Tang, and Juanzi Li. A unified approach to researcher profiling. *Proceedings of 2007 IEEE/WIC/ACM International Conferences on Web Intelligence*, 2007.

[43] Steve Young. Using pomdps for dialog management. *IEEE/ACL Workshop on Spoken Language Technology (SLT 2006), Aruba*, 2006.

[44] Stefan Ziesemer. *Namenserkennung bekannter und unbekannter Namen*. 2007.