Adaptive Speech Understanding for Intuitive Model-based Spoken Dialogues

Tobias Heinroth, Maximilian Grotz, Florian Nothdurft, Wolfgang Minker

Institute of Communications Engineering – Dialogue Systems
University of Ulm, Albert-Einstein-Allee 43, D-89081 Ulm
prename.surname@uni-ulm.de

Abstract

In this paper we present three approaches towards adaptive speech understanding. The target system is a model-based Adaptive Spoken Dialogue Manager, the OwlSpeak ASDM. We enhanced this system in order to properly react on non-understandings in real-life situations where intuitive communication is required. OwlSpeak provides a model-based spoken interface to an Intelligent Environment depending on and adapting to the current context. It utilises a set of ontologies used as dialogue models that can be combined dynamically during runtime. Besides the benefits the system showed in practice, real-life evaluations also conveyed some limitations of the model-based approach. Since it is unfeasible to model all variations of the communication between the user and the system beforehand, various situations where the system did not correctly understand the user input have been observed. Thus we present three enhancements towards a more sophisticated use of the ontology-based dialogue models and show how grammars may dynamically be adapted in order to understand intuitive user utterances. The evaluation of our approaches revealed the incorporation of a lexical-semantic knowledgebase into the recognition process to be the most promising approach.

Keywords: Dialogue management, Failure prevention, Keyword spotting

1. Introduction

Within the framework of the EU-funded project ATRACO we have conducted a series of evaluation sessions to primarily find out how users cope with adaptive and "intelligent" systems, residing within Intelligent Environments (IEs) (Goumopoulos and Kameas, 2009). The ATRACO system provides several user interfaces that adapt to devices and services, which are dynamically integrated into the system. One of these interfaces is a Spoken Dialogue System (SDS) that is managed by the OwlSpeak Spoken Dialogue Manager (SDM) (Heinroth et al., 2010). OwlSpeak generates dialogue descriptions (e.g., in VoiceXML) on the fly depending on the current context. Here *context* refers to the status of the ATRACO system and thus to the status of the IE with respect to the current user task (i.e., when relaxing, studying, cooking, etc.) and to the input of the user. The SDM utilises ontologies as multi-domain dialogue descriptions for specific devices, services, or for information retrieval. It combines these descriptions during run-time into a multi-purpose spoken dialogue interface.

In (Heinroth et al., 2010) we presented details about the model-based OwlSpeak SDM. Model-based approaches towards SDM yield considerable advantages as they clearly separate domain-dependent and domain-independent knowledge. Such a separation offers many adaptation capabilities, and, due to the predefined models, also provides robust dialogues. Compared to statistical approaches (e.g., (Young, 2007)) where costly corpora incorporating all contingencies are mandatory, the OwlSpeak SDM is able to render the dialogue context dynamically including the required dialogue aspects on the fly. However, the followed approach lacks in its capacities when it comes to unforeseen situations that are not described appropriately within the pre-defined models. How can a more flexible and therefore adaptive behaviour of the model-based SDM be obtained in order to achieve an *intuitive* spoken interface? In this paper we present several methods that may be used to exploit a dialogue model more intelligently. We also show how semantic data may enrich the model during run-time by querying a lexical-semantic net (GermaNet, see (Hamp and Feldweg, 1997)).

The following section provides an overview on related work. In Section 3 we provide some insights into the results of a real-life ATRACO project evaluation that motivated our work. In Section 4 we provide details on the implemented prototype and show how the SDM benefits from the different methods. The setup of the evaluation is described in Section 5. Section 6 provides the results of the evaluation that has been carried out. The paper concludes and provides an outlook on future work in the last section.

2. Related Work

Established SDSs that, for example, have been implemented by means of frameworks such as TrindiKit (Larsson and Traum, 2000) or Olympus (Bohus et al., 2007) have demonstrated a high performance within specific (predefined) domains such as bus line information or flight booking. However, when it comes to real-life spoken interaction within likely changing domains that evolve or even may be substituted by other domains, it seems to be problematic to utilize these heavy-weight approaches appropriately. Furthermore, when it comes to intuitive dialogue situations nowadays SDSs show severe limitations.

Since speech is not a "crisp" communication channel difficulties arise when a computer interprets spoken user input (McTear et al., 2005). Such difficulties can usually be ascribed to misinterpretations caused by the recognizer, for example, when it is not possible to map an audio signal to a word that is part of the applied grammar. We use the term *grammar* for a formal definition of the input provided by the recognizer. This definition can be correlated with semantic meanings that in turn can be evaluated by the SDM.

One way of deciding if an audio signal can be mapped to the grammar is to calculate a confidence measure (Jiang, 2005). To detect if a successful mapping can be assumed, a predefined threshold has to be exceeded. We talk about a *non-understanding* if an input cannot successfully be mapped to a term defined as part of the grammar. In contrast, and independent from a contextual and semantic correctness, the successful mapping of an input to a term is referred to as *understanding*. Obviously, it makes sense to avoid the occurrence of non-understandings. However, solving this issue by implementing huge grammars that cover nearly all possible inputs would not be beneficial as this inevitably leads to more *misunderstandings* (i.e., false positives). In such a case the user would need to proactively correct the input and repair the dialogue, which is costly and cognitively demanding. In that sense, a non-understanding would even be beneficial since it allows the system to query the user to confirm an input that could not be understood at the first attempt. Thus the size of the grammar is a trade-off between understanding too few or too much input.

As for many other applications the key is to steer a middle course. One option is to step-wise broaden the grammar during a second or third process of recognition (Chung et al., 2004). Herewith the grammar can be extended depending on the context. This approach is relevant to our work since the ontologies used in OwlSpeak as spoken dialogue models are perfectly suited to be extended during runtime. A further interesting approach that is relevant to our work is a methodology called ISSS (Incremental Significant utterance-Sequence Search). It follows the idea of analysing an input step-by-step without initially knowing the whole input (Nakano et al., 1999). The authors tried to recognise the input on a word-by-word basis and built up a knowledgebase consisting of several possible input variations. This knowledgebase is actualised for each newly recognised word. Once an end of the input has been detected the most appropriate system reaction is selected and provided to the user.

A further approach has been proposed in the recent past: a second level of recognition (López-Cózar and Callejas, 2006). Here the first level comprises of a comprehensive grammar covering all possible inputs the user may utter within the application domain. The first recognition level provides a graph of words. This graph is a network constituted of words (corresponding to the nodes) and probability transitions between the words (corresponding to the arcs). The second level of recognition comprises an analysis of the graph of words. Three parameters are important for the analysis: a set of word classes (consisting of keywords), the current prompt the SDS uttered before, and the transition probabilities. The authors showed that their approach significantly enhanced the recognition accuracy compared to a similar SDS that utilises a prompt-dependent grammar. A limitation of the proposed technique is that the recognition enhancement does not involve the decision logic of the SDM. This means, for example, that the system is not able to filter out utterances that do not have a (semantic) meaning within the domain. The approach is relevant to our work since we apply a second level of recognition as well. In the following section we present the motivation of our work.

3. Motivation

In order to cope with an evolving dialogue domain the OwlSpeak SDM proposes a flexible handling of different dialogue domains. We have designed several lightweight ontologies that are utilised as dialogue models. These ontologies can be activated, deactivated, and most notably combined with each other during run-time (i.e., during the ongoing dialogue). Hence the system is able to adapt to changing domains that especially occur within IEs. However, initial real-life studies within the framework of ATRACO revealed a main lack of the model-based approach towards SDM: Even the most sophisticated predefined model cannot cover all contingencies. Figure 1 shows the number of non-understandings that occurred during three comprehensive evaluation sessions conducted with six test persons. Each session took between 22 and 60 minutes.

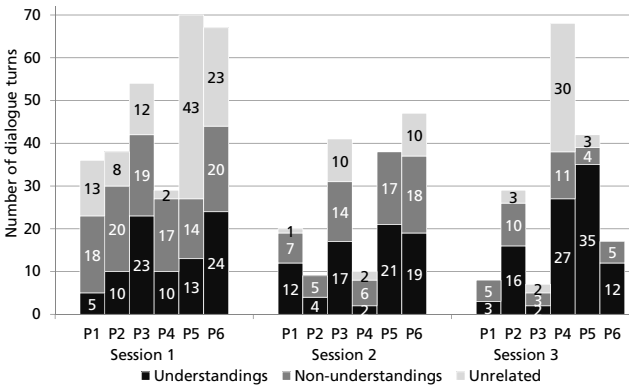

Figure 1: The number of understandings, non-understandings, and unrelated utterances (van Helvert et al., 2010).

In order to make the numbers comparable we have normalised them to 30 minutes. It seems to be obvious that we received a high number of non-understandings on average. In total 212 non-understandings and 252 understandings have been recorded. Furthermore the system detected 162 unrelated utterances that the system rejected correctly. Even though these numbers are quite sobering, the system has to be robust against interjecting utterances (e.g., "Oh, my god") since we applied an "always listening" setup. The main challenge is to find ways to utilise the spoken dialogue models describing a specific domain in order to properly react on non-understandings. In case the user input does not match any of the grammatical expressions that are predefined as part of the dialogue model, the system should be able to detect the most probable command.

Thus, in this paper we present our attempt to overcome the drawbacks of the approach while keeping the benefits that have been discussed in (Heinroth and Denich, 2011). In the following we propose three approaches that lead to fewer non-understandings without increasing the number of misunderstandings. As explained in the introduction the OwlSpeak SDM uses on-the-fly generated VoiceXML Documents to describe a currently active dialogue. These dialogues are newly generated every three to five seconds,

depending on the setup of the SDM. It is not possible to manipulate a grammar of a VoiceXML document while the interpreter parses the document. This is the main reason why we propose a nested analysis of the user input after the grammar has failed (Rohlicek et al., 1989).

OwlSpeak uses a set of ontologies for various domains as dialogue models. Usually a dedicated ontology per device or service is implemented. We have extended these ontologies by a set of keywords that can be used to detect the actual domain (i.e., the topic of the user input). Thus, an ontology that describes the possible (spoken) commands that can be used to control, for example, a lamp as part of an IE, could provide keywords such as "light", "lamp", "shiner", and "luminary". These can then be used by the nested recogniser to find out to which domain the utterance may belong to. After the system has detected to which ontology the utterance possibly refers to, a second analysis is started by utilising keywords that are specific for a concrete command within the detected domain. Regarding the previous example such keywords could be "on", "off", "bright", or "low". If this analysis concludes with a valid result, i.e., "domain=light" and "command=low" the SDM generates a confirmation dialogue and asks the user if he wants the light to be low. This could be answered by "yes" or "no". On the one hand this procedure avoids a second user input that could result in a non-understanding again. On the other hand the system does not end up performing the wrong command automatically because of a misleading combination of keywords. In the following we call this approach *keyword-based*.

An alternative mode of recognition does not make use of a fixed (limited) grammar but of an extensive dictionary. We refer to this as out-of-vocabulary recognition (OOV). As a result the system would receive, for example, an n-best list of results from the recognizer (cf. Listing 1). Obviously, due to the lack of a grammar, this can easily lead to confusion. There is, for example, a high probability that two similar-sounding words, such as "house" and "mouse" may be mistaken. A way to grasp the user's intention is to determine the underlying semantic meaning of the utterance. This could be done by a string-based comparison between the n-best list and the words listed in the grammar (which has not led to an understanding before). However, this would lead to a significant number of mistakenly detected non-understandings. In other words, if the grammar lists the word "light" an OOV recognition would provide an n-best list containing {might, flight, right,...}. The SDM would still (after the second analysis) only be able to emit a non-understanding. Obviously, especially homophones are problematic within this context. To encounter this, we propose to blur the grammar by adopting the Levenshtein distance (Levenshtein, 1966). It is described as the minimal number of insertions, deletions, or substitutions of characters needed to transform one string into another. In order to detect homophones that have erroneously been recognized the Levenshtein distance of the recognized word and the words that are part of the grammar can be calculated pair-wise. For the German language a distance of one would be sufficient. For the English language there are several homophones with a higher distance (e.g., "colonel" and "kernel"), which would admittedly lead to more confusions. Thus we propose to adopt a low distance in order to benefit from the blurred grammar without producing too many misunderstandings. In the following we refer to this approach as *blurred-keyword*.

```
<?xml version="1.0" encoding="utf-8"?>
<results>
 <result confidence=0,21>
        Aber Fernseher aus</result>
 <result confidence=0,21>
        aber Fernseher aus</result>
 <result confidence=0,17>
        Aber Fernsehen aus</result>
 <result confidence=0,19>
        aber Fernsehen aus</result>
 <result confidence=0,18>
        warum aber Fernsehr aus</result>
 <result confidence=0,17>
        aber fern der aus</result>
 <result confidence=0,16>
        aber fern sehr aus</result>
 <result confidence=0,20>
        am Anfang sehr aus</result>
 <result confidence=0,168>
        aber wenn der aus</result>
 <result confidence=0,17>
        Amor Fernseher aus</result>
</results>
```

Listing 1: The n-best list provided by the OOV recogniser.

A further attempt that utilises OOV recognition is the semantic interpretation of the user input. A dialogue model for handling a greeting situation may provide "hello", "good morning" or other greeting forms. However, if a specific greeting form such as "hi" is not covered by the grammar (note that we utilise a minimal grammar in order to reduce overlapping grammars and misunderstandings), the system will not be able to react appropriately. Thus, we propose to figure out the semantic meaning of "hi", which can then be mapped on the semantic value "salutation". This value is encoded within the dialogue model to be the semantic meaning of, for example, "hello". We utilise GermaNet, a German lexical-semantic dictionary as an external semantic knowledgebase (Hamp and Feldweg, 1997). GermaNet provides relations to detect synonyms, hyponyms, and hypernyms. The proposed mechanism may also be used to dynamically broaden the grammar of the dialogue model and thereby extend the model during runtime. We refer to this method as *semantic-keyword*. In the following we present the implementation of the three methods before we present the results of the evaluation.

4. Implementation

The presented work has been implemented as part of the OwlSpeak SDM framework. The aim is to enhance the recognition capabilities. We have modified the generation of the VoiceXML documents to allow a transfer of the recorded user utterance to an external recognizer (after a non-understanding occurs). For that purpose we use the Microsoft Speech API (MS SAPI). The API allows to use a grammar (i.e., a list of keywords) or to perform an OOV recognition based on the MS SAPI language model. The transfer of the user utterance to the recognizer is performed by the built-in VoiceXML variable "application.lastresult$.recording", which is then passed as a wav-file to the external recogniser. If the input can be successfully analysed, the result is passed back to the OwlSpeak SDM. It reacts appropriately by modifying the dialogue

model and generating a new VoiceXML document. Figure 2 shows a flow-diagram of the keyword-based approach that detects key utterances that are related to a specific dialogue model. The approach allows, for example, if the system fails in recognizing a user input such as “lights *background noise* on” in the first attempt, to correctly detect “lights on”. For this purpose the system detects “lights”, which is provided as an ontology keyword by the domain-specific dialogue model (Figure 2-1). Afterwards a list of command-keywords for the specific domain is used to detect the word “on” (Figure 2-2). Upon successful processing, a confirmation dialogue is automatically generated and the dialogue can proceed. If the system does not detect a matching input a third recognition is performed using the command-keywords of the last dialogue that was actually involved within the interaction (Figure 2-3).

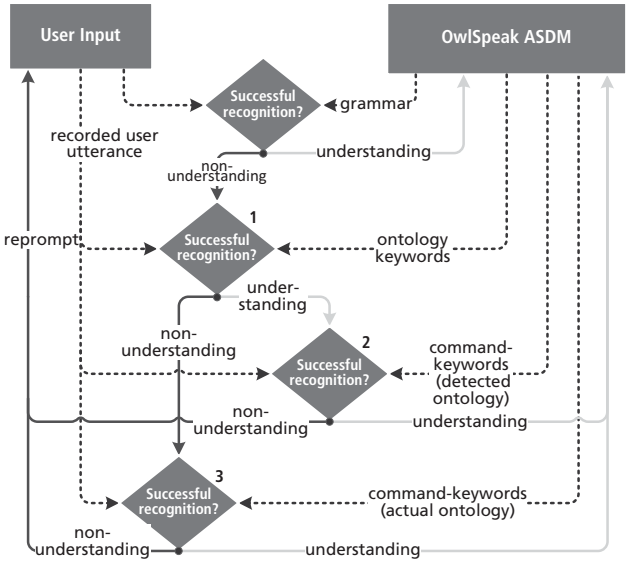

Figure 2: Flow-diagram of the keyword-based approach.

Figure 3 depicts the blurred-keyword approach using the Levenshtein-Distance. In case of a non-understanding detected by the regular recognizer, the system performs OOV recognition to be able to understand the user input. The result, an n-best list, is used to pair-wise calculate the Levenshtein distance for the same keywords we have already utilised as part of the keyword-based approach. During a first step (Figure 3-1) the domain related keywords are used. Usually German homophones have a maximum Levenshtein distance of one. Hence we accept a distance that is lower than two to indicate an understanding. During a second step (Figure 3-2) this calculation is repeated utilising the command-specific keywords. In case of a further understanding we continue with a confirmation dialogue. Analogously to the keyword-based approach we perform a third analysis of the user input by calculating the distance of the n-best list and the command-related keywords of the last ontology that has been involved in dialogue generation (Figure 3-3). The default behaviour of the system is invoked in case the distance of this last analysis is greater or equal two: the system repeats the last prompt (if a question has to be answered) or it behaves passively.

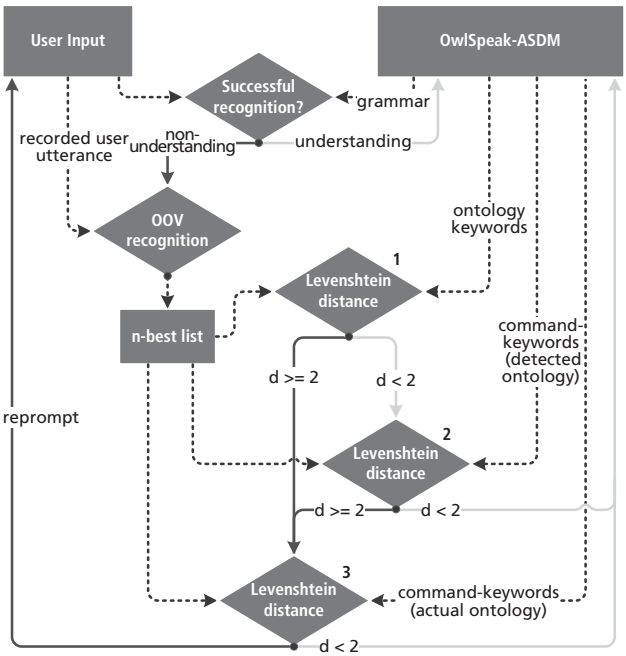

Figure 3: Flow-diagram of the blurred-keyword approach.

An entirely different problem is a corresponding input using words that are not covered by the grammar and therefore cannot be detected by neither the blurred-keyword nor the keyword-based approach. Figure 4 shows the semantic-keyword approach incorporating a semantic knowledge-base to analyse the user’s intention.

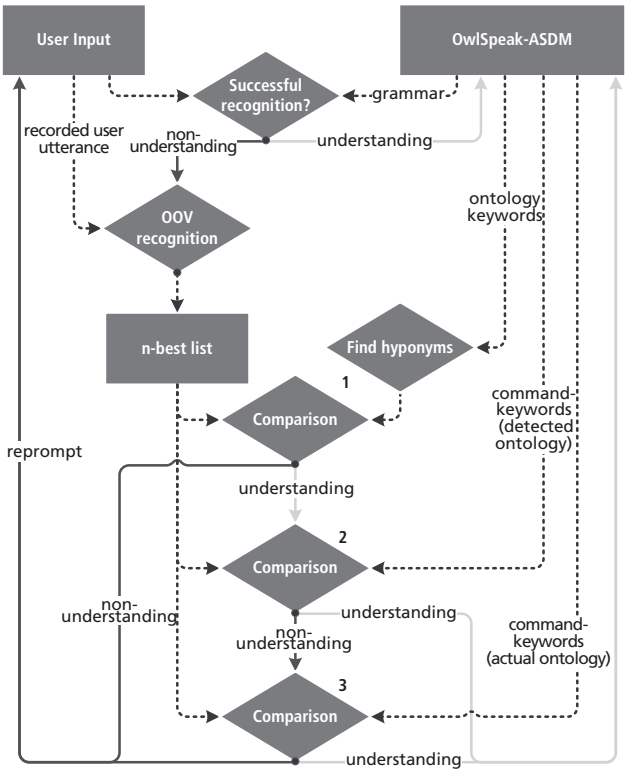

Figure 4: Flow-diagram of the semantic-keyword approach.

After the regular recognizer detects a non-understanding, the n-best list provided by the OOV recognition is semantically analysed using the GermaNet API (Figure 4-1). The system uses the hyponyms of all ontology keywords and pair-wise compares them in order to find semantic similarities. As mentioned above, all ontologies provide domain-related keywords. In other words, the system would detect "lamp" or "torchiere" for the ontology keyword "ceiling light". In case of a positive match the system checks if any command-related keywords occur within the corresponding entry of the n-best list (Figure 4-2). In case of a non-understanding, the system proceeds with the dialogue and ignores the last user input. The subsequent comparison is processed analogously to the keyword-based approach (Figure 4-3). For these two steps we have disregarded from an optional comparison based on hyponyms since GermaNet does not provide verbs and only a few adverbs. In case of a positive match the system asks the user to verify the utterance. If the user confirms, the SDM is able to carry out the command or to process the new information.

We assume that the model-based OwlSpeak SDM benefits most from our approach when it comes to spontaneous and intuitive user-system communication. A realistic use-case for our system would be a command-based SDS used by non-experts. Such a use-case requires an intuitive handling of the voice interface. In the following section we focus on the setup of the evaluation before we present the results of the test series that has been carried out in order to proof the capabilities of our system.

5. Evaluation Setup

For the evaluation we decided to implement a scenario related to home automation. Figure 5 shows the virtual room that provided a visual feedback for the commands the subjects had to utter *intuitively*. Since fostering the intuitiveness of the OwlSpeak-based SDS was a main motivation of our approaches, we did not reveal the subjects the commands the SDS is actually able to understand. Thereby all subjects were forced to control the virtual environment as they personally assumed. The OwlSpeak system delivered minimal grammars for the six devices within the environment. Of course, if a user is aware of the possible commands the SDS works appropriately with these grammars. However, this is not what we intended to evaluate. Instead the main aim was to evaluate how users intuitively cope with such a system. As depicted in Figure 5, a ventilator, a TV, a stereo, the heating, a jalousie, and a lamp could be controlled. The subjects may use commands such as "Switch the light on!" or "Volume up". These commands were part of the original ontologies describing the dialogues. For the evaluation we have enriched these dialogue models with keywords for all domains and for the corresponding commands. For the stereo domain we introduced keywords such as "audio equipment", "hi-fi", or "music". The SDS that has been used consists of the OwlSpeak SDM, a Voxeo Prophecy 10 speech platform, a Loquendo Speech Suite 7 (including TTS, ASR, and MRCP), a SIP-Client (Linphone), and the Microsoft Speech API (grammar-based + OOV) utilised as external recogniser.

Figure 5: Virtual room used to visualise the test-bed.

The virtual room provided visual feedback encoded by using the colours red, blue, and green. Deactivated devices are coloured red, activated devices are coloured green, and blue devices are currently changing their state, e.g., the volume of the device changes. The explanation of the virtual room was part of the short introduction the subjects received. The subjects' goal was simple and comprehensive in unison: They have been told to control the environment. 40 subjects took part of the evaluation: 30 male and ten female users. In order to get an idea of the subjective user estimation the subjects had to fill in a tailored SASSI questionnaire (Hone and Graham, 2000) on a scale from zero (strongly disagree) to seven (strongly agree) regarding the following measures:

EFF: Efficiency shows how efficient the system is, i.e., how well the dialogue flow can be followed by the user.

REL: Reliability shows how reliable the system is regarding to mistakes and understanding problems.

FRI: Friendliness describes how user-friendly the system is and how pleasant it is for the subject to interact with the system.

All test persons were at least fluent German speakers and the system therefore was implemented as a German SDS. In order to allow a comparison of the three recognition enhancements with the baseline system we divided the 40 subjects into four groups each consisting of 10 people. Each group had to use the virtual test-bed by freely controlling the various devices. We terminated a test run after approximately 22 commands depending on the duration (15 minutes). In total 900 spoken commands have been recorded. The most important objective metrics we have investigated are the number of understandings, non-understandings, and misunderstandings. Given a pure command-and-control system, metrics such as task-completion or dialogue success rate are less relevant. In order to avoid misunderstandings that may occur, we have implemented a confirmation dialogue that is initiated whenever one of our enhancements

detects a spoken command that has not been covered by the grammar. An example of such an automatically generated dialogue is presented in Table 1. In the following section we present the subjective and the objective results of the evaluation and investigate how the results influence our research.

	Utterance	Reaction
User	Please, the TV, could you switch it on?	-
System	-	[Keyword1 = TV] [Keyword2 = on]
System	Do you want to switch the TV on?	-
User	Yes.	-

Table 1: A confirmation dialogue the system generates automatically.

6. Evaluation Results

Figure 6 shows the subjective user estimations for all groups on average. The approaches are measured neutrally (4) with slight deviations. The average values show a tendency that Group B who used the *keyword-based* approach rated the system worse than the subjects of all other groups. Regarding EFF (efficiency) and REL (reliability) these differences are significant. The Kruskal-Wallis Test calculates a P value below 0.05 for both measures.

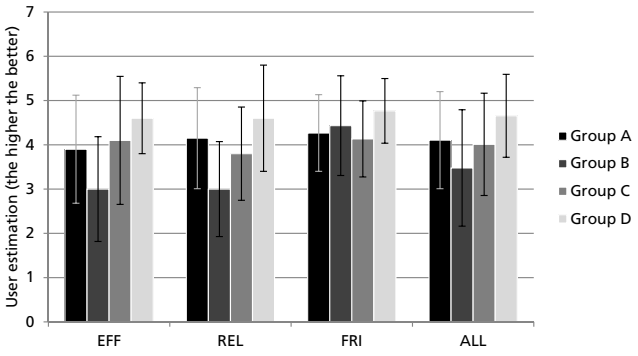

Figure 6: The subjective results of the questionnaire on average together with the standard deviation.

The results indicate that regarding EFF and REL the *semantic-keyword* approach outperforms the other methods. Regarding the subjective estimation of user-friendliness we did not receive a comparable significant result. The last subjective measure (ALL) gives an indication of the overall system estimation. This group of bars summarises the subjective rating for all users: The *keyword-based* approach was rated worst, the *baseline* and the *blurred-keyword* approach were rated neutrally, and the *semantic-keyword* approach is rated best. Again, these results are significant: The Kruskal-Wallis Test calculates an exact significance of P = 0.04. During an initial test phase 10 subjects used the system without any enhancements (Group A). In the following this session is regarded as *baseline*. In total, the Group A users uttered 223 commands. Only 68 commands (31%) were correctly understood by the system. Of course, this bad result was expected and conditioned by the setup of the evaluation: the users had to *intuitively* use the spoken command system.

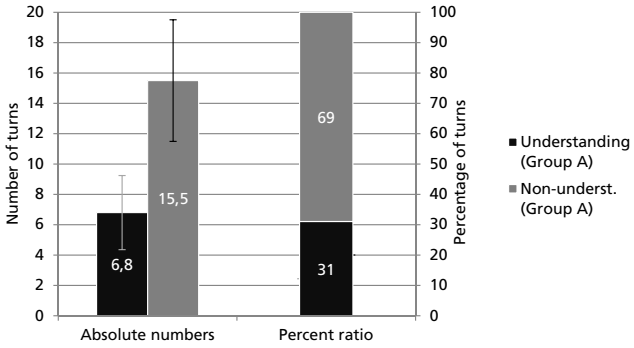

Figure 7: The average numbers and the percentages of understandings and non-understandings that occurred during the Group A session.

The grammar covered 33% of the spoken input during the *keyword-based* session, 26% during the *blurred-keyword* test run, and 30% during the evaluation of the *semantic-keyword* session. These numbers underpin that even a simple spoken command-and-control system cannot to be intuitively used. Figure 8 shows the objective results of the keyword-based approach. In total 219 utterances have been recorded. 73 (33%) of these utterances have been covered by the grammar. The additional keyword-spotter correctly detected 83 (37%) utterances.

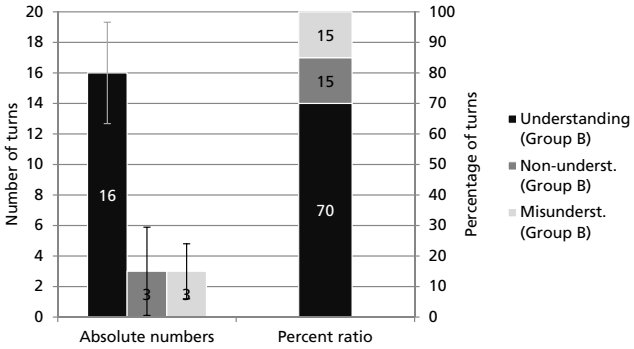

Figure 8: The average numbers and the percentages of understandings, non-understandings, and misunderstandings that occurred during the Group B session.

Hence, the system was able to understand 70% of all intuitively uttered commands. However, we also received 32 (15%) non-understandings and 31 (15%) misunderstandings in total. The misunderstandings were the main reason for the bad subjective rating of the keyword-based approach. Compared to the *baseline* approach the improvement is significant. The Mann-Whitney U-Test shows a P value of 0.008 disproving the hypothesis that the two results are statistically equal. However, this improvement strongly depends on the quality of the pre-defined keywords. For a

scaled-up version of the system we assume that the task of selecting appropriate keywords is nothing but trivial. Figure 9 shows the results of the blurred-keyword approach. 221 commands have been recorded during this evaluation session. Only 26% of these utterances have been covered by the built-in grammar. The blurred-keywords improved this rate by 51% eventuating in a total recognition rate of 77% (163 correctly understood commands). A difference between the keyword-based approach and the blurred-keyword setup was the usage of OOV recognition. We suppose this kind of recognition avoided the occurrence of misunderstandings. Hence, we believe in a scaled up scenario the blurred-keyword approach may be beneficial due to the higher user input coverage of the blurred keywords. Compared to the *baseline* approach the *blurred-keywords* improved the understanding rate by 46%. Again, the U-Test shows a very low P value of 0.004 underpinning the significance of the improvement.

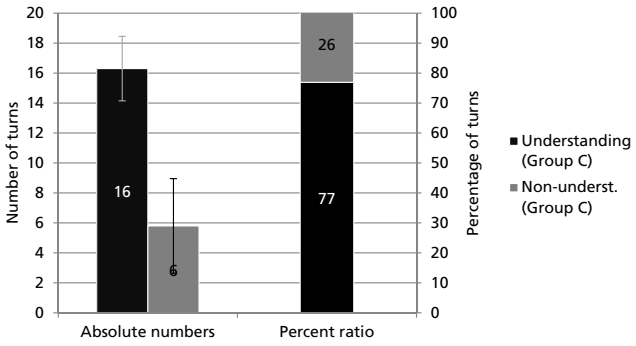

Figure 9: The average numbers and the percentages of understandings and non-understandings that occurred during the Group C session.

Figure 10 depicts the results of the semantic-keyword approach. During this test run the grammar covered 30% of the 237 commands that have been spotted. The combination of the semantic-lexical knowledgebase and the predefined keywords led to 123 additional utterances that were correctly recognised (52%). Especially within a larger dialogue domain we estimate the usage of the OOV recogniser and the dynamic extension of keywords to be beneficial.

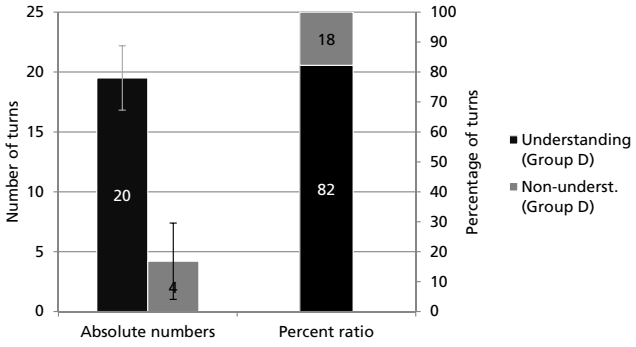

Figure 10: The average numbers and the percentages of understandings and non-understandings that occurred during the Group D session.

The better rating of the semantic-keyword approach regarding the user-friendliness also indicates that the subjects felt more comfortable since GermaNet allowed a broader usage of terms leading to a more natural behaviour in practice. This benefit is important especially with respect to colloquial utterances, e.g., "box" for television set. As with the previous session the U-Test shows a very low P value of 0.002. This underpins the significance of the improvement between the grammar-based recognition and the *semantic-keyword* approach.
Such naming of devices that should be controlled via voice rarely occurs. However, a system able to understand this naming is perceived to be more natural, which in this case relates to more user-friendliness. Table 2 shows the naming of the six devices the subjects used in German.

Stereoanlage (stereo)	**64**	**Fernseher (telly)**	**141**
Musik	51	TV	6
Radio	41	Fernseh	6
Anlage	11	Fernsehgerät	4
Musikanlage	4	TV-Gerät	3
Radioreceiver	2	Fernsehen	1
Lautsprecher	1	Glotze	1
Audioanlage	1		
CD-Player	1		
Radiolautsprecher	1		
Jalousie (sun-blinds)	**62**	**Licht (lights)**	**91**
Rolladen	58	Lampe	56
Rollo	25	Stehlampe	12
Vorhang	1	Leuchte	2
Fenster	1	Stehleuchte	1
Ventilator (fan)	**88**	**Heizung (heating)**	**139**
Lüfter	9	Heizkörper	8
Gebläse	2	Radiator	3
Lüftung	1	Wärmequelle	1

Table 2: The naming of the devices the subjects used and the frequency the subjects used them.

Obviously, the most common identifiers (marked as bold with their English translation) are most frequently used. However, several uncommon names have rarely been used (e.g., "Audioanlage" for the English word "stereo"). It would be hard to develop a grammar that covers such a variety of names for a higher number of devices and services. A further problem of large grammars is their maintenance. A main criterion how to choose the various commands a large-scale grammar consists of, is that the utterances should not sound similar in order to avoid misunderstandings. Lightweight grammars can fulfil this important requirement due to their lower complexity. In the following section we summarise the approach and draw some conclusions before taking a look at future work.

7. Conclusion

In this paper we have presented three approaches that have been implemented in order to enhance the understanding capabilities of an OwlSpeak-based SDS. On the one hand the presented approaches avoid the necessity of commands

to be repeated by the user. On the other hand they avoid misunderstandings by generating confirmative questions. By using keywords and the segmentation of the input it is possible to understand the intention of the user without comprehending the whole utterance.

We have integrated keywords on the domain level and on the command level. These keywords can be utilised by the system to analyse a user utterance in case the built-in lightweight grammars fail matching the spoken input. During the evaluation within the IE domain the simple keyword-based approach showed good results. However, as indicated by the subjective user estimation, a main drawback of this approach was that several misunderstandings occurred. These are very disruptive to SDSs. Within a larger domain we assume that such a keyword-based approach would perform worse since the method strongly depends on the quality of the pre-defined keywords. Thus we have proposed two more intelligent ways of handling non-understandings that usually arise during a spoken dialogue. The blurred-keyword approach utilises an OOV recognizer and analyses the utterance by comparing the recognised n-best list with the keywords. The approach matches within a specific Levenshtein distance. The approach runs without any misunderstandings and performs slightly better than the keyword-based method.

Due to the capabilities of the OOV recogniser the blurred-keywords cover more variations of the pre-defined keywords, thus making the approach more flexible. However, the third method, the semantic-keyword approach, performs better. The subjects also indicated the semantic-keyword detection to be the most user-friendly one. By utilising the GermaNet semantic-lexical knowledgebase and OOV recognition it outperforms the keyword-based approach regarding the number of positive matches. Hence we assume this approach to be the most suitable extension for model-based SDS. We estimate that it would achieve similar results within larger domains with more commands and more keywords. The OOV recognition combined with the semantic knowledgebase is capable to adaptively provide a meaningful basis for the recognition especially within situations where users intuitively interact with an SDS. In the future we are planning to conduct a further test session with expert users who are aware of the commands the baseline system is able to understand. We assume that even expert users will only slightly outperform the intuitively used semantic-keyword approach.

Acknowledgements

The research leading to these results has received funding from the Transregional Collaborative Research Centre SFB/TRR 62 "Companion-Technology for Cognitive Technical Systems" funded by the German Research Foundation (DFG).

8. References

Dan Bohus, Antoine Raux, Thomas K. Harris, Maxine Eskenazi, and Er I. Rudnicky. 2007. Olympus: an open-source framework for conversational spoken language interface research. In *Workshop on bridging the gap: Academic and industrial research in dialog technologies*, pages 32–39. Association for Computational Linguistics.

G. Chung, S. Seneff, C. Wang, and L. Hetherington. 2004. A dynamic vocabulary spoken dialogue interface. In *International Conference on Spoken Language Processing (ICSLP)*, pages 1457–1460.

C. Goumopoulos and A. Kameas. 2009. Ambient ecologies in smart homes. *The Computer Journal*, 52(8):922–937.

B. Hamp and H. Feldweg. 1997. Germanet – a lexical-semantic net for german. In *ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 9–15.

Tobias Heinroth and Dan Denich. 2011. Spoken Interaction within the Computed World: Evaluation of a Multitasking Adaptive Spoken Dialogue System. In *COMPSAC 2011*, Munich, Germany. IEEE.

Tobias Heinroth, Dan Denich, Alexander Schmitt, and Wolfgang Minker. 2010. Efficient spoken dialogue domain representation and interpretation. In *LREC'10*, Valletta, Malta. ELRA.

Kate S. Hone and Robert Graham. 2000. Towards a tool for the Subjective Assessment of Speech System Interfaces (SASSI). *Natural Language Engineering*, 6:287–303.

Hui Jiang. 2005. Confidence measures for speech recognition: A survey. *Speech Communication*, 45(4):455–470.

Staffan Larsson and David Traum. 2000. Information state and dialogue management in the trindi dialogue move engine. *Natural Language Engineering*, pages 323–340.

V.I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet Physics Doklady*, volume 10, pages 707–710.

R. López-Cózar and Z. Callejas. 2006. Two-level speech recognition to enhance the performance of spoken dialogue systems. *Knowledge-Based Systems*, 19(3):153–163.

M. McTear, I. O'Neill, P. Hanna, and X. Liu. 2005. Handling errors and determining confirmation strategies–an object-based approach. *Speech Communication*, 45(3):249–269.

Mikio Nakano, Noboru Miyazaki, Jun-ichi Hirasawa, Kohji Dohsaka, and Takeshi Kawabata. 1999. Understanding unsegmented user utterances in real-time spoken dialogue systems. In *The 37th annual meeting of the Association for Computational Linguistics*, pages 200–207, Stroudsburg, PA, USA. Association for Computational Linguistics.

JR Rohlicek, W. Russell, S. Roukos, and H. Gish. 1989. Continuous hidden Markov modeling for speaker-independent word spotting. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP'89)*, pages 627–630. IEEE.

Joy van Helvert, Hani Hagras, and Achilles Kameas. 2010. D27 - prototype testing and validation. Technical report, ATRACO Project (FP7/2007-2013).

S. Young. 2007. Using POMDPs for dialog management. In *IEEE Spoken Language Technology Workshop*, pages 8–13. IEEE.