A Voiceportal Enhanced by Semantic Processing and Affect Awareness

Felix Burkhardt, Joachim Stegmann, Markus Van Ballegooy

T-Systems International GmbH

(felix.burkhardt|joachim.stegmann|markus.van-ballegooy)@t-systems.com

Abstract: In order to improve the automation rate of state-of-the-art IVR systems we introduce the application of two key technologies. Semantic processing is performed to analyze the input utterance of the user and to generate a suitable system output based on a given domain model. Additionally, affect awareness is used for monitoring the emotional state of the user.

1 Introduction

This paper proposes a possible enhancement of IVR (interactive voice response: systems that communicate via voice-interface) systems using technologies based on semantic processing and affect awareness. The main objective is to overcome the limitations of fixed call-flow design using open-ended prompts and modeling of limited domains based on semantic nets. Using these technologies, the automation rate of complex business processes (e.g. in consulting hotlines) can be increased.

On the other hand, limitations of automation caused by human factors have to be taken into account. In case of anger or stress of the user, special dialog strategies or, alternatively, a transfer to a human agent have to be applied in order to reduce the number of lost calls.

The article is structured as follows. The next section describes the system architecture. Two following sections focus on the key technologies, namely semantic processing and anger detection. A last section elaborates on the conciliation strategies that we used in a first pilot study.

2 System Architecture

The voice platform consists of a standard voice browser supporting VoiceXML 2.0 ([voi]) (see figure 1). The application server is based on the Java 2 Enterprise Edition (J2EE) standard and contains all the application logic and the modules for affect awareness and semantic processing. The speech dialog is controlled by the interaction manager.

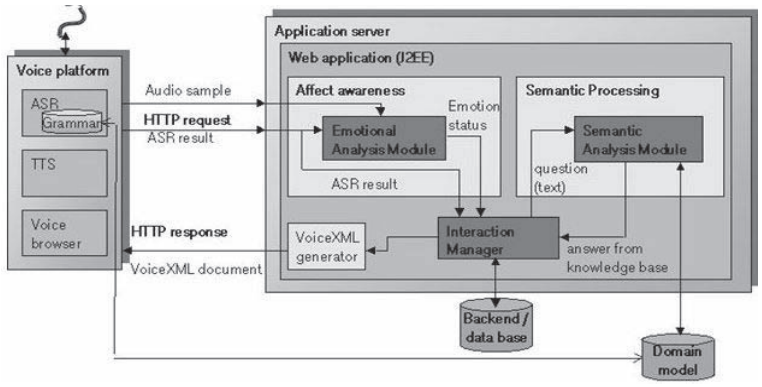

Figure 1: Architectural overview

The interaction manager receives the ASR result and forwards this text string to the semantic processing unit, where the text is analyzed. The output of the semantic processing unit is a text string containing the answer for the user or a set of refinement candidates. For the first case, the audio prompt for output to the user is dynamically built from a database of basic audio segments. In the latter case the interaction manager asks the user for further refinement of his request. For progressing to the next dialog step VoiceXML is generated dynamically. During all processing steps the output of the emotional analyzer is monitored by the interaction manager. If anger is detected in the caller's voice, the interaction manager runs a conciliation strategy by first mirroring the caller's mood and in a second step transferring the caller to a human agent.

3 Semantic Processing

An intuitive way to incorporate semantics in an IVR system is to integrate it in the speech recognition grammar. This means, that the application designer has to map any possible user utterance to a given category or entry in the speech menu. This procedure is suitable for a limited and simply structured domain, however, for complex and wide-spread domains it can be advantageous to separate the speech recognizer from the semantic processing. Using this approach, the speech recognizer performs the speech-to-text conversion of a user utterance and transmits a text string to the semantic processing unit for further interpretation. In a second step, the text string is analyzed based on a given domain model. These analyzing steps include question analysis, interpretation and answer generation.

The workflow of the semantic processing unit is shown in figure 2. Semantic analysis of the user input is based on a knowledge base, whose domain is described by ontology. The ontology is modeled using the open-source software Protégé [NFN01]. Additionally, we use a semantic lexicon which contains a large variety of expressions with their related synonyms and a list of frequently asked questions (FAQ) for the given domain.

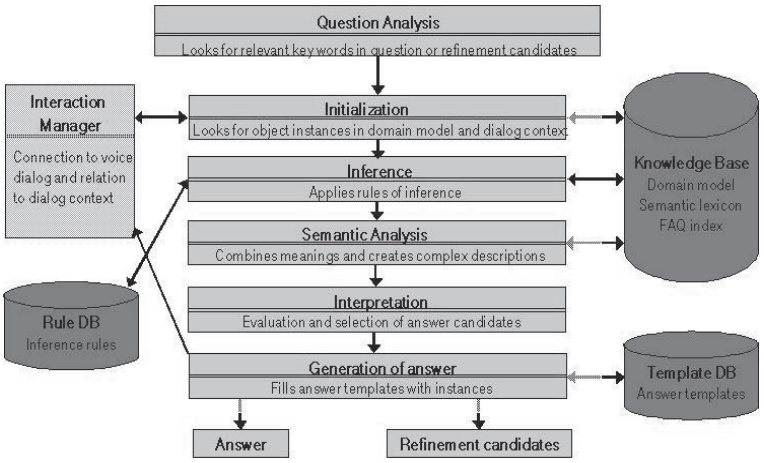

Figure 2: Workflow semantic processing

The system can handle grammatically incorrect and incomplete sentences (from speech input) and can be easily adapted to the domain. The question analysis module looks for relevant key words or refinement candidates in the question. It is based on the SProUT system from DFKI [DKP+04]. The next processing step in Figure 2 is an initialization of the basic terms for the identified keywords. This initialization is based on the semantic lexicon. After that, the system checks if the user question can directly be answered using the FAQ list of the given domain. If this is not the case, rules of inference are applied to introduce new concepts and solve ambiguities.

The semantic analysis combines concepts and returns complex descriptions of the answer candidates using relations and descriptions of the concepts given in the ontology. If the user question is not precise enough the system recognizes that more details are necessary to generate an answer. In this case, a number of refinement candidates are returned to the interaction manager.

If there is more than one valid answer for the given question the answer candidates have to be evaluated and ranked using a heuristic approach. If all answer candidates are equal in ranking, all of them are used in the answer. Finally, the answer is dynamically generated from the concepts describing the relevant answer candidates using a large database of pre-recorded audio segments.

4 Emotion Recognition

A channel of human communication that is not considered in today's automated voice systems is the emotional expression. If customers get frustrated by talking to a machine that might not directly understand the user's requests, there is no way to express this anger

and the caller feels not understood even more. As unsatisfied customers can become quite expensive, industry is motivated to find ways to tackle this problem with automated strategies. As a key system component for handling this problem, we propose to integrate an emotional recognizer in a state-of-the-art IVR system.

The acoustic anger detection is handled by the statistical classifier is described in the article *"Enhancing a Voiceportal Using Classification of Emotion with Acoustic Features"* by R. Huber, F. Gallwitz and V. Warnke, this issue. It analyses prosodic features like fundamental frequency, energy and derivatives of them. The training has been performed based on a labeled database with three hours of recorded conversations.

First evaluation conducted on faked (acted) data show that an overall recognition (equal error rate) of about 75 % can be expected. However, by using thresholds, i.e. only regard the value of either class, the result can be tweaked in favour of either neutral or anger detection and a correctness of 98 % for neutral utterances be achieved (see figure 3).

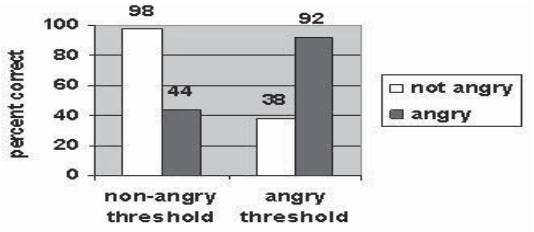

Figure 3: Anger detection results tweaked by thresholds (left:non-anger>0.96, right: anger>0.95)

5 Conciliation Strategies

While transferring conciliation strategies from the inter-human communication into dialog behavior of an emotional-aware voice portal we face two major constraints:

1. A machine might be able to detect anger, but it is yet ignorant of the anger's causation. To be credible, only those strategies can be transferred into man-machine dialogs that are generic enough so that it is not required to make references to the content subject of the negative emotion.

2. Conciliation strategies in a dialog system must be designed very straightforward and narrow so that the user will not be encouraged to stray from the task .

Despite these constraints we tried to transfer two conciliation strategies into the speech dialog of our pilot study. It was set up as a self service application where customers can look up their telephone bills and find information about their mobile phone tariffs.

1. Within dialog situations where slight anger was detected and where there were no

further hints that the users request could not be handled within the voice portal, we used a strategy that is called ”conciliation by mirroring”. The goal of this strategy is to show the user that his emotions are recognized but that it is better to continue the task. Immediately after an utterance that was classified as angry by the acoustic classifier, our system interrupted the main dialog by playing a short prompt like: ”*I see that you are a little bit excited. So it will be the best if I continue quickly with your query!*”

2. Within dialog situations where strong anger was detected in combination with hints that the dialog will not be completed successfully we used a strategy called ”conciliation by empathy and delegation” by offering the user the possibility to leave the voice portal and speak to a ”real” service agent. Before the connection was put through, the system verbally showed empathy for the users situation. We used a prompt like: ”*I notice that you are angry because I don't grasp your request. I can really understand you! The best thing will be if I put you through to a service agent.*”

6 Conclusions

We described a voice portal system that includes semantic modeling and the detection of anger in the caller's voice. It is based on a standard VoiceXML framework and can easily be integrated into existing architectures.

7 Acknowledgments

Most of the work described was carried out in the framework of a project funded by Deutsche Telekom, Zentralbereich Innovation and partially funded by the EU NoE HUMAINE.

References

[DKP+04] Witold Drozdzynski, Hans-Ulrich Krieger, Jakub Piskorski, Ulrich Schäfer, and Feiyu Xu. Shallow Processing with Unification and Typed Feature Structures — Foundations and Applications. *Künstliche Intelligenz*, 1:17–23, 2004.

[NFN01] S. Decker M. Crubezy R. W. Fergerson M. A. Musen N. F. Noy, M. Sintek. Creating Semantic Web Contents with Protege-2000. In *IEEE Intelligent Systems*, volume 16(2), pages 60–71, 2001.

[voi] Voice Extensible Markup Language (VoiceXML) Version 2.0. http://www.w3.org/TR/voicexml20.