WITcHCRafT: A Workbench for Intelligent exploraTion of Human ComputeR conversaTions

Alexander Schmitt[1], Gregor Bertrand[1], Tobias Heinroth[1], Wolfgang Minker[1], Jackson Liscombe[2]

[1]Institute for Information Technology, University of Ulm, Germany
[2]SpeechCycle Inc., New York, Broadway 26, USA
{alexander.schmitt,gregor.bertrand,tobias.heinroth, wolfgang.minker}@uni-ulm.de,
jackson@speechcycle.com

Abstract

We present Witchcraft, an open-source framework for the evaluation of prediction models for spoken dialogue systems based on interaction logs and audio recordings. The use of Witchcraft is two fold: first, it provides an adaptable user interface to easily manage and browse thousands of logged dialogues (e.g. calls). Second, with help of the underlying models and the connected machine learning framework RapidMiner the workbench is able to display at each dialogue turn the probability of the task being completed based on the dialogue history. It estimates the emotional state, gender and age of the user. While browsing through a logged conversation, the user can directly observe the prediction result of the models at each dialogue step. By that, Witchcraft allows for spotting problematic dialogue situations and demonstrates where the current system and the prediction models have design flaws. Witchcraft will be made publically available to the community and will be deployed as open-source project.

1. Introduction

The growing task complexity of spoken dialogue systems such as telephone-based speech applications requires new tools to support system designers and speech scientists in analyzing human machine dialogues. By finding problematic and critical situations within the human-machine "conversation" an iterative improvement of the service can be facilitated. By "critical" we mean in the context of automated telephone applications, situations where the caller is about to hang up without completing the task because he is annoyed by the automation in general or stuck and helpless by faulty system behavior. Especially in longer lasting dialogues such as in automated technical support agents where dialogues frequently consist of more than 50 system and user turns, one quickly loses track of how the conversation between the user and the system happened. Listening to recorded calls for debugging and troubleshooting is far too time consuming and does not allow for an overall view of the conversation.

On the other hand, modern speech dialogue platforms allow for extensive logging during the conversation between user and system and provide various parameters. Lacking expressiveness when stored in "flat" databases, these huge amounts of data can be brought into a form, where they are readable and interpretable for researchers, system developers and call center agents. At this point, the Witchcraft workbench comes into play, bringing logged conversations back to life.

2. Related Work

Extensive work on the prediction of problematic dialogue situations in human-machine conversations has been carried out by Walker et al. (Walker et al., 2002). By "problematic" they denote dialogues that failed, i.e. where no solution was reached in collaboration between user and the automated system. Walker et al. employ a rule-learning algorithm, to implement a prediction model forecasting the outcome of calls in the HMIHY (How May I Help You) call routing system from AT&T. Their classifier is able to distinguish between "problematic" and "non-problematic" calls and is trained with logged interaction data from problematic and non-problematic calls. Problematic calls are transferred to an operator completing the task jointly with the caller. Similar studies can be found in (Levin and Pieraccini, 2006) and (Paek and Horvitz, 2004). In earlier work we presented a task completion predictor for automated agents providing telephone-based technical support (Herm et al., 2008). Trained on the employed dialogue corpus, it also predicts "problematic" calls, i.e. calls where the task has not been completed. Closely related to such interaction log-based predictors are acoustic classifiers for detecting anger (Lee and Narayanan, 2005) and other information about the user such as age and gender (Metze et al., 2007). Our anger classifier as described in (Schmitt et al., 2009) is able to determine angry user turns based on acoustic and contextual information.

All studies in this context consider "only" corpus-related performance. At latest when deployment of such models is scheduled, their direct influence on *specific* dialogues should be of highest interest to system developers and researchers building such models. Note, that we use the terms "predictors" and "classifiers" in this context interchangeably.

Going live and deploying such classifiers is a critical point since it can have a severe and also negative impact on the dialogue. In our point of view an analysis of the classifier's impact on specific calls is mandatory. This and the huge amount of data we are dealing with motivated us in implementing Witchcraft.

3. Application Scenarios

We call Witchcraft a "workbench" since it may be used for a variety of applications centered on the evaluation of

human computer dialogues and the analysis of prediction and classification models. Although Witchcraft has been designed for telephone-based speech applications it may also be used for an analysis of dialogues logged by any other dialogue system since the underlying concepts remain the same. Witchcraft is first of all not an annotation or transcription tool in contrast to other workbenches such as Transcriber (Geoffrois et al., 2000), the MATE workbench (McKelvie et al., 2001) or DialogueView[1].
The role of Witchcraft is to support the *analysis of logged dialogues* plus the identification of problems and to support the *analysis of prediction and classification models* whose purpose is to render dialogue systems more "intelligent". For example Witchcraft is able to

- estimate, at each dialogue step, the likelihood that the user will successfully end the task, e.g. as proposed by (Walker et al., 2002), (Levin and Pieraccini, 2006) and (Kim, 2007)
- estimate the emotional state, e.g. (Lee and Narayanan, 2005), and other information about the user such as age and gender, e.g. (Metze et al., 2007).

3.1. Call Browsing

How a deployed system performs is hardly traceable in larger speech-applications with long-lasting dialogues. Listening to recorded calls is too time-consuming and does not provide an insight into the logged parameters and the dialogue flow. Witchcraft presents the complete dialogue containing system prompts, system actions, ASR accuracy, recognized word strings, parsed semantics, the call reason etc. in a structured and easily accessible manner at each dialogue step. The Witchcraft user can jump into any position within the dialogue and start replaying. System prompts are synthesized with a text-to-speech (TTS) engine but also the pre-recorded prompt from the professional speaker could be used for prompting. The user utterances are played back from original and logged recorded conversations.

3.2. Model Analysis

Task completion models help to detect problematic situations in ongoing calls and allow for repair strategies or a transfer to an operator. Similarly, anger, age and gender classification models help to render a dialogue more robust and natural.
It is not visible how the model would act online in a deployed application. How should we know, if the model mistakenly causes a transfer to an operator due to false classification? Our workbench allows visualizing the impact of employing such models and shows, for each dialogue turn, possible predictions of the specific model at any given point in time. By that, we can deduce, for example, when:

- the task completion prediction model recommends a transfer to a human agent
- the system "thinks" the user is angry based on the anger classification model, see Figure 1
- the system seems to be certain that it is talking to a male/female or junior/senior person based on gender and age models.

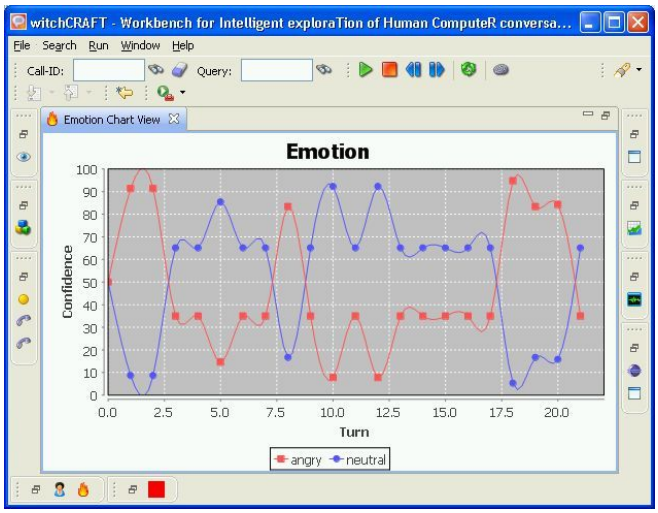

Figure 1: Turnwise emotion detection model applied on a call originating from an automated troubleshooter IVR system being able to resolve Internet-related problems. The red line symbolizes the confidence of the classifier for the caller being angry in that turn. The blue line is the confidence for a neutral utterance (0-100%).

4. Application Overview and Implementation

4.1. Overview

Figure 2 presents the *Analysis Perspective* whose central component is the call browser. When a call is selected, statistics such as average ASR confidence, call reason, outcome (whether the call was escalated or not), number of turns, etc. are displayed in the *Call Detail View*. Furthermore, gender, emotion (angry/annoyed vs. neutral), age and task completion prediction are estimated for each dialogue turn and plotted in the chart views.
In datasets with several ten thousand calls, orientation is highly important. Witchcraft features a grouping functionality within the *Call Selection View*, allowing to group calls by their unique IDs into *IDGroups* or by a database query (in SQL) into *QueryGroups*. Thereby, calls with specific properties can be looked up from the database and analyzed with Witchcraft. Examples for such QueryGroups are: calls from male callers, calls from people who lost their password, calls which lasted at most 1 minute. IDGroups make sense, when interesting calls are selected for further investigation during the datamining process. By that, e.g. IDs from angry callers can be stored in one group.

4.2. Implementation

To implement Witchcraft, we have chosen Java since it allows for platform independency and integration of existent Java libraries and frameworks, such as chart libraries, TTS engines and RapidMiner (Mierswa et al., 2006), a powerful machine learning framework, for model building and

[1] http://cslu.cse.ogi.edu/DialogueView/

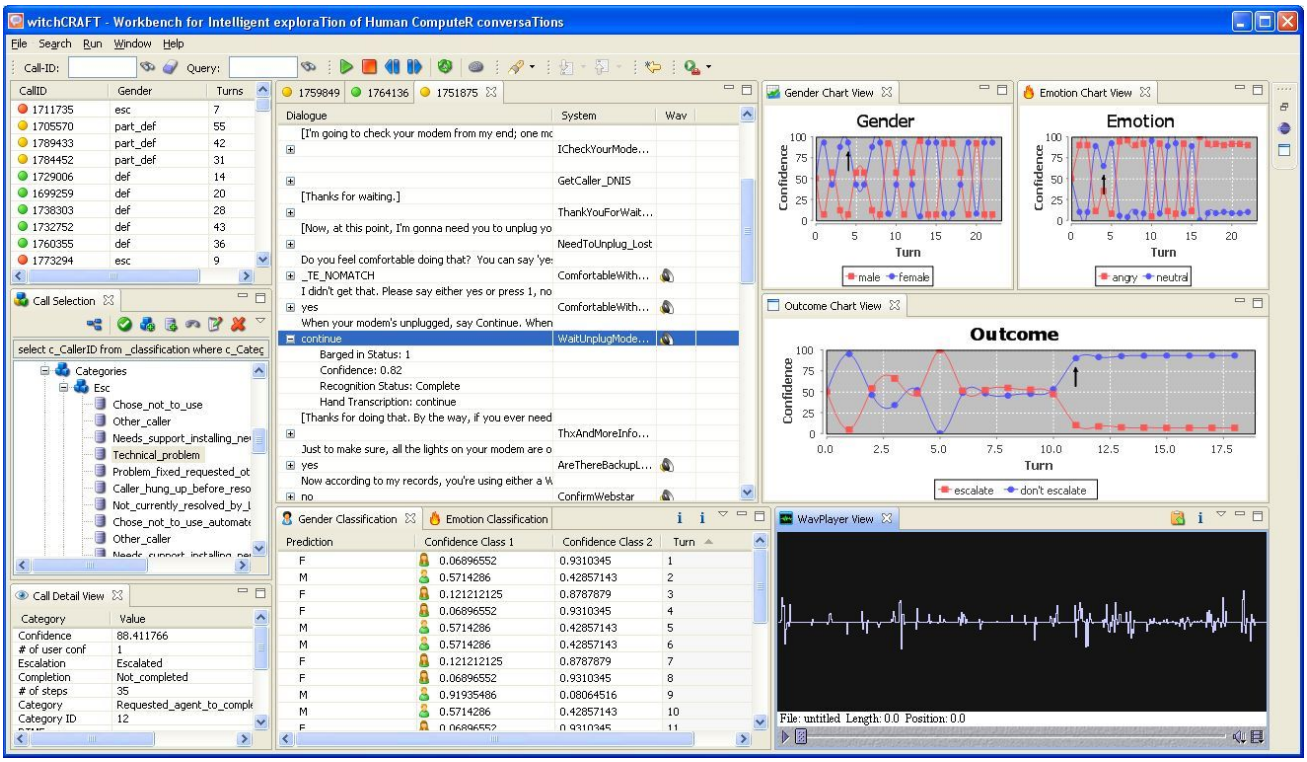

Figure 2: Screenshot of the Witchcraft Workbench in the Analysis Perspective. It consists of Call Selection Views, Classification Tables and Chart Views, the player for concatenated user utterances and the central Dialogue View enabling a navigation within the conversation.

classification. For an optimum flexibility within the user interface, we applied the Eclipse RCP framework[2]. The core of the popular integrated development environment Eclipse enables an organization of the user interface in various views and perspectives. In Witchcraft we have developed two main perspectives: the *Analysis Perspective* containing prediction charts, details on the dialogue turns etc. and a *Conversation Perspective* (see Figure 3) allowing for an overview of the conversation. New perspectives can be defined by grouping different existing components on the screen. This allows for a strong adaptation to specific tasks. The employed component architecture allows for the development of third-party plug-ins and components for Witchcraft without the need for getting into detail of the existing code. This facilitates the extension of the workbench by other developers.

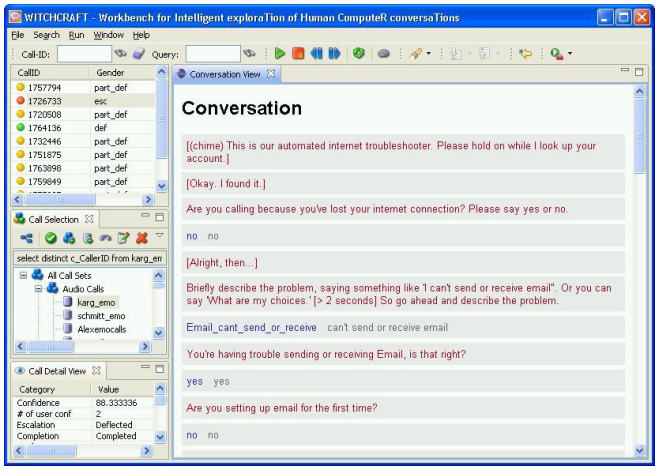

Figure 3: Conversation Perspective displaying a call. The red prompts stem from the IVR system, black prompts are automatic transcriptions of the user utterances and grey text according hand transcriptions.

5. Prediction and Classification Models

5.1. Model Training

To get a notion, where those prediction models applied in Witchcraft originate from, we describe the process of model training by means of our domain (see Figure 4). We trained various prediction and classification models with RapidMiner on collected data from a technical support automated agent resolving internet problems.

The data comprises interaction log information such as ASR transcription, confidence, semantic parse, number of reprompts etc. that have been captured during the conversation in the database. How problematic and non-problematic calls are defined, is up to the domain. In our approach, we assigned calls that have been transferred to an operator the label problematic, i.e. the classifier predicts the escalation. Calls, which have been successfully completed, were labeled as non-problematic. For training anger-, age- and gender models the recorded user utterances have been manually labeled with anger, age and gender labels. They were subject to a feature extraction process with Praat (Boersma and Weenink, 2009) delivering prosodic and acoustic fea-

[2]www.eclipse.org/home/categories/rcp.php

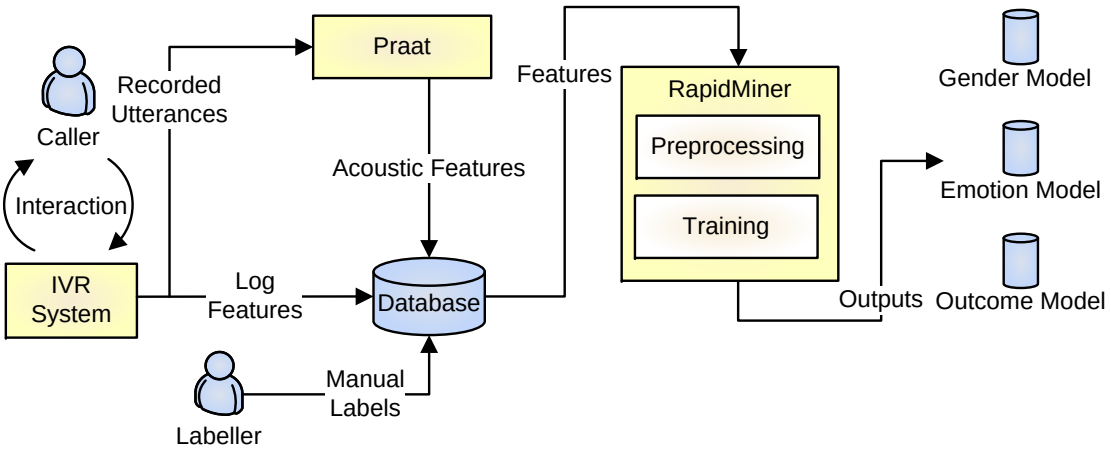

Figure 4: Typical procedure of preparing and training prediction/classification models prior to applying them on specific calls with Witchcraft.

tures. Labels and features are then subject to the training with RapidMiner. Preprocessing is performed to remove strongly correlated features.

5.2. Model Application

The delivered models can either be batch-tested within RapidMiner resulting in overall corpus specific performance (e.g. accuracy, f-score etc). Batch-testing is the typical procedure for an end-to-end evaluation. However, they models may now also been evaluated with Witchcraft where they are applied on specific calls. The integration of the prediction models in Witchcraft is shown in Figure 5. Witchcraft can instruct the integrated RapidMiner component to apply the available prediction and classification models on the currently analyzed call. The effects of applying the models are then visualized in the different chart views. Please note that Witchcraft is currently only able to interface and use RapidMiner as classifier. Additional modules interfacing MATLAB[3] or Weka (Witten and Frank, 2005) are envisioned but not yet realized. RapidMiner, however, provides an exhausting set of machine learners, including all Weka learners and can thus be considered as good choice.

6. Conclusion and Discussion

Witchcraft turned out to be a valuable tool in everyday work when dealing with thousands of dialogues and analyzing prediction models. In detail the workbench provides the following features:

- huge dialogue corpora are accessible and manageable
- decisive information of a dialogue are presented at one glance
- the conversation between system and user can be entirely simulated based on logs
- information about the dialogue and the speaker, such as task completion probability at each turn, anger status, gender and age by using underlying prediction models are displayed
- the evaluation of such prediction models is now possible on the dialogue level instead of the corpus level
- it will be open-source and easily extendable by writing new Java-based EclipseRCP components

There are still some restrictions that have to be clarified prior to deployment:

- *proprietary identifiers:* the underlying database uses proprietary identifiers for the interaction parameters that are sometimes specific to the domain Witchcraft initially was designed for. Our current work thus targets on removing any proprietary information and on generalizing identifiers to make an employment of Witchcraft for other researchers in other dialogue domains as uncomplicated as possible. This includes a definition of a taxonomy for interaction parameters that will be published separately.
- *sample data:* the current setup is based on a corpus that is not publically available. To demonstrate Witchcraft we are currently adapting a freely available corpus that will be deployed together with Witchcraft as an out-of-the-box solution.

Currently, an employment of Witchraft in new domains requires the following adaptations:

- Interaction logs need to be brought into the SQL-based database format Witchcraft uses.
- For making use of the model testing capabilities of Witchcraft domain-dependent prediction models need to be trained according to Figure 4.

Future work will also include the development of a search mechanism that will allow for directly searching problematic dialogues with Witchcraft and an integration of

[3] www.mathworks.com

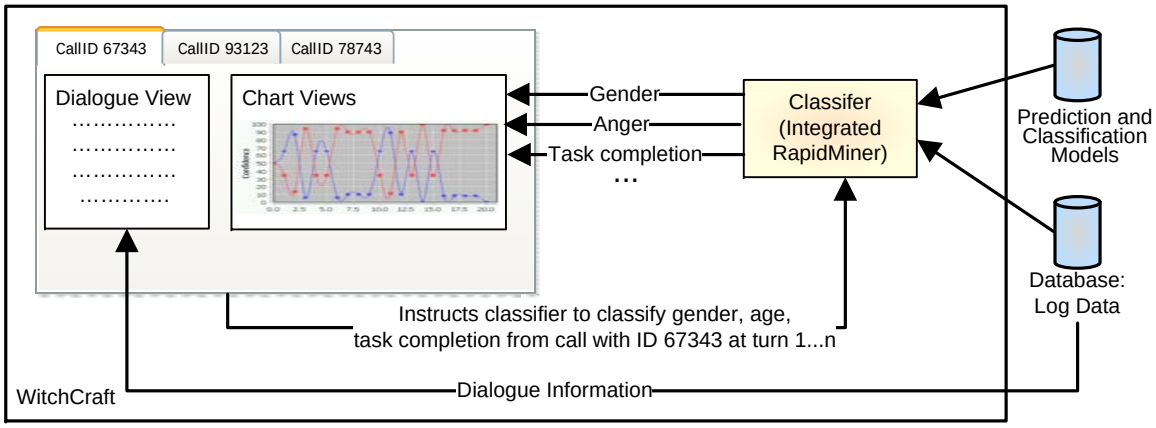

Figure 5: Application of prediction models on specific dialogues: Witchcraft requests turn-wise predictions from the classifier and displays the estimations in the chart views.

Weka and MATLAB as classifier engines. We are planning to make Witchcraft freely available to the community after leaving the beta-status. It will be hosted under GNU General Public License at Sourceforge under witchcraftwb.sourceforge.org.

7. 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).

The authors would like to thank the reviewers for their positive and valuable comments and all students an co-workers involved in the development of Witchcraft.

8. References

Paul Boersma and David Weenink. 2009. Praat: doing phonetics by computer (version 5.1.04), April.

Edouard Geoffrois, Claude Barras, Steven Bird, and Zhibiao Wu. 2000. Transcribing with annotation graphs. In *Proceedings of the Second International Conference on Language Resources and Evaluation*, pages 1517–1521, Paris. ELRA. EN.

Ota Herm, Alexander Schmitt, and Jackson Liscombe. 2008. When calls go wrong: How to detect problematic calls based on log-files and emotions? In *Proc. of the International Conference on Speech and Language Processing (ICSLP) Interspeech 2008*, pages 463–466, September.

Woosung Kim. 2007. Online call quality monitoring for automating agent-based call centers. In *Proc. of the International Conference on Speech and Language Processing (ICSLP)*, September.

Chul Min Lee and S. S. Narayanan. 2005. Toward detecting emotions in spoken dialogs. *IEEE Transactions on Speech and Audio Processing*, 13(2):293–303, March.

Esther Levin and Roberto Pieraccini. 2006. Value-based optimal decision for dialog systems. In *Proc. of Spoken Language Technology Workshop 2006*, pages 198–201, Dec.

David McKelvie, Amy Isard, Andreas Mengel, Morten Baun Møller, Michael Grosse, and Marion Klein. 2001. The mate workbench - an annotation tool for xml coded speech corpora. *Speech Communication*, 33(1-2):97 – 112. Speech Annotation and Corpus Tools.

Florian Metze, Jitendra Ajmera, Roman Englert, Udo Bub, Felix Burkhardt, Joachim Stegmann, Christian Müller, Richard Huber, Bernt Andrassy, Josef Bauer, and Bernhard Littel. 2007. Comparison of four approaches to age and gender recognition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1.

Ingo Mierswa, Michael Wurst, Ralf Klinkenberg, Martin Scholz, and Timm Euler. 2006. Yale: Rapid prototyping for complex data mining tasks. In Lyle Ungar, Mark Craven, Dimitrios Gunopulos, and Tina Eliassi-Rad, editors, *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 935–940, New York, NY, USA, August. ACM.

Tim Paek and Eric Horvitz. 2004. Optimizing automated call routing by integrating spoken dialog models with queuing models. In *HLT-NAACL*, pages 41–48.

Alexander Schmitt, Tobias Heinroth, and Jackson Liscombe. 2009. On nomatchs, noinputs and bargeins: Do non-acoustic features support anger detection? In *Proceedings of the 10th Annual SIGDIAL Meeting on Discourse and Dialogue, SigDial Conference 2009*, London (UK), September. Association for Computational Linguistics.

Marilyn Walker, I Langkilde-Geary, H W Hastie, J Wright, and A Gorin. 2002. Automatically training a problematic dialogue predictor for a spoken dialogue system. *Journal of Artificial Intelligence Research*, (16):293–319.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition edition.