Advances in the Witchcraft Workbench Project

Alexander Schmitt, Wolfgang Minker
Institute for Information Technology
University of Ulm, Germany
alexander.schmitt,
wolfgang.minker@uni-ulm.de

Nada Ahmed Hamed Sharaf
German University in Cairo, Egypt
nada.sharaf@student.guc.edu.eg

Abstract

The *Workbench for Intelligent exploraTion of Human ComputeR conversaTions* is a new platform-independent open-source workbench designed for the analysis, mining and management of large spoken dialogue system corpora. What makes Witchcraft unique is its ability to visualize the effect of classification and prediction models on ongoing system-user interactions. Witchcraft is now able to handle predictions from binary and multi-class discriminative classifiers as well as regression models. The new XML interface allows a visualization of predictions stemming from any kind of Machine Learning (ML) framework. We adapted the widespread CMU Let's Go corpus to demonstrate Witchcraft.

1 Introduction

Substantial effort has been invested in the past years in exploring ways to render Spoken Dialogue Systems (SDS) more adaptive, natural and user friendly. Recent studies investigated the recognition of and adaption to specific user groups, e.g. the novices and expert users, or the elderly (Bocklet et al., 2008). Further, there is a massive effort on recognizing angry users, differentiate between genders (Burkhardt et al., 2007), spotting dialects, estimating the cooperativeness of users or user satisfaction (Engelbrecht et al., 2009) and finally, predicting task completion (Walker et al., 2002). When applied *online*, i.e. during the interaction between user and system, these models can add valuable information to the dialogue system which would allow for an adaption of the dialogue strategy, see Figure 1.

Until now we can report that these models[1] work more or less well in batch-test scenarios offline. An anger classifier might deliver 74% accuracy when evaluated on utterance level. But which impact would the deployment of this recognizer have on specific dialogues when being employed in a real system? Would it fail or would it succeed? Similarly, at what point in time would models predicting gender, speaker age, and expert status deliver a reliable statement that can indeed be used for adapting the dialogue? What we need prior to deployment is an evaluation of the models and a statement on how well the models would work when being shifted on dialogue level. At this point, the Witchcraft Workbench enters the stage.

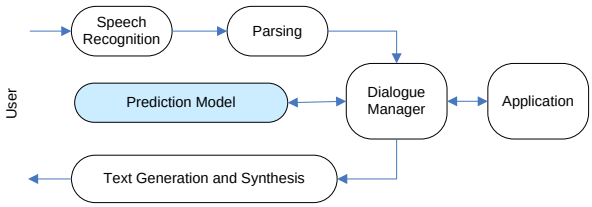

Figure 1: Enhanced SDS: The prediction model that is used to render the dialogue system more user-friendly delivers additional information to the dialogue manager.

2 The Role of Witchcraft

For a more detailed introduction on the Witchcraft Workbench please refer to (Schmitt et al., 2010a). In a nutshell, Witchcraft allows managing, mining and analyzing large dialogue corpora. It brings logged conversations back to life in such that it simulates the interaction between user and system based on system logs and audio recordings. Witchcraft is first of all not an annotation or transcription tool in contrast to other workbenches such as NITE (Bernsen et al., 2002), Transcriber[2]

[1] please note that we use the expression recognizer, classifier and prediction model interchanging in this context

[2] http://trans.sourceforge.net

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 261–264,
The University of Tokyo, September 24-25, 2010.

or DialogueView[3]. Although we also employ it for annotation, its central purpose is a different one:Witchcraft contrasts dialogue flows of specific dialogues which are obtained from a dialogue corpus with the estimations of arbitrary prediction and classification models. By that it is instantly visible which knowledge the dialogue system would have at what point in time in the dialogue. Imagine a dialogue system would be endowed with an anger recognizer, a gender recognizer and a recognizer that should predict the outcome of a dialogue, i.e. task completion. Each of the three recognizers would be designed to deliver an estimation at each point in the dialogue. How likely is the user angry? How likely is he male or female and how likely will the task be completed based on what we have seen so far in the dialogue. To which extent the recognizers deliver a correct result can be verified within Witchcraft.

3 Handling Models in Witchcraft

Witchcraft had several shortcomings when we first reported on it in (Schmitt et al., 2010a). It was only working with a proprietary industrial corpus and was heavily tailored to our needs. It worked only with specific models from binary discriminative classifiers. Since then we have put substantial effort to generalize the functionality and to make it available to the community.

To allow an analysis of other recognizers the system has been extended to further handle predictions from multiclass discriminative classification and regression tasks. Witchcraft does not contain "intelligence" on its own but makes use of and manages the predictions of recognizers. We assume that a recognizer is implemented either as stand-alone recognizer or with help of a Machine Learning (ML) framework. We emphasize that Witchcraft itself does neither perform feature extraction nor classification. Witchcraft operates on turn level requesting the recognizer to deliver a prediction based on information available at the currently processed dialogue turn of a specific dialogue. Where and how the recognizer accomplishes this is not part of the architecture. The ML framework of our choice that was originally supported natively, i.e. directly accessed by Witchcraft (Schmitt et al., 2010a) was RapidMiner[4], an ML framework that covers a vast majority of supervised and unsupervised machine learning techniques. The initial plan to interface other ML frameworks natively (such as MatLab, the R framework, BoosTexter, Ripper, HTK that are frequently used in research) turned out not to be practical. In order to still be able to cover the broadest possible range of ML tools we introduced a new generic XML interface. For simplicity we removed the RapidMiner interface. An overview of the dependency between Witchcraft and a recognizer is depicted in Figure 2.

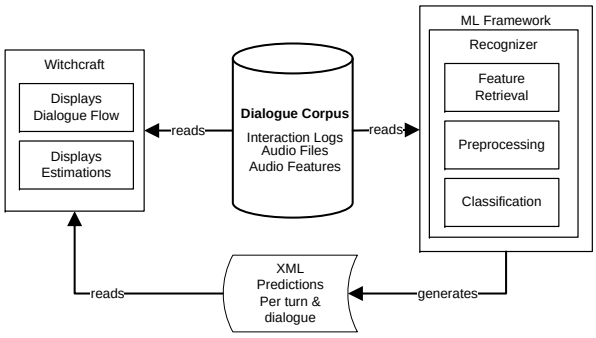

Figure 2: Dependency of Witchcraft and related recognizers that are implemented within an ML framework.

Witchcraft has been extended to support an arbitrary number of models, see Figure 3. They can now be one of the types "discriminative binary", "discriminative multiclass classification" and "regression".

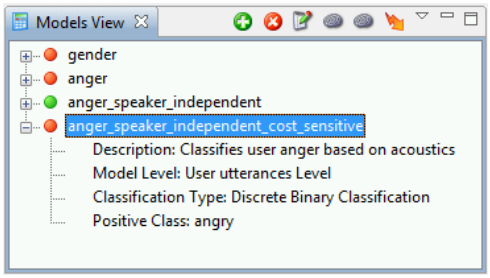

Figure 3: Definition of a model within Witchcraft. External recognizers have to deliver predictions for the defined models as XML documents.

A recognizer implemented in an ML framework has to be defined in such a way that it delivers XML documents that fit the model definition in Witchcraft. Each XML document represents the prediction of the recognizer for a specific dialogue turn of a specific dialogue. It contains for discriminative classification tasks, such as gender, or emotion the number of the turn that has been classified,

[3]http://cslu.cse.ogi.edu/DialogueView/
[4]www.rapid-i.net

the actual class label and the confidence scores of the classifier.

```
<xml>
<turn>
<number>1</number>
<label>anger</label>
<prediction>non-anger</prediction>
<confidence class='anger'>0.08</confidence>
<confidence class='no-ang'>0.92</confidence>
</turn>
</xml>
```

In regression tasks, such as the prediction of user satisfaction, retrieving cooperativeness scores etc., the returned result contains the turn number, the actual label and the prediction of the classifier:

```
<xml>
<turn>
<number>1</number>
<label>5</label>
<prediction>3.4</prediction>
</turn>
</xml>
```

After performing recognition on a number of dialogues with the recognizer Witchcraft reads in the XML files and creates statistics based on the predictions and calculates dialogue-wise *accuracy, f-score, precision and recall values, root mean squared error etc.* The values give some indication of how precisely the classifier worked on dialogue level. That followed it allows to search for dialogues with a low overall prediction accuracy, or e.g. dialogues with high true positive rates, high or low class-wise f-scores etc. via SQL. Now a detailed analysis of the recognizer's performance on dialogue level and possible reasons for the failure can be spotted.

4 Evaluating Models

In Figure 4 we see prediction series of two recognizers that have been applied on a specific dialogue: a gender recognizer that predicts the gender on turn basis and an emotion recognizer that predicts the user's emotional state (angry vs. non-angry) at the current turn. The red line symbolizes the confidence of the recognizers for each of the predicted classes. For example, in the emotion model the blue line is the confidence for a non-angry utterance (0-100%), the red line for an angry one. Exemplary for the two models we take a closer look at the gender model. It predicts the gender on turn basis, i.e. it takes the current speech sample and delivers estimations on the speaker's gender. As we can see, there are a number of misrecognitions in this call. It stems from a female speaker but the recognizer frequently estimated a male speaker. The call could be spotted by searching within Witchcraft for calls that yield a low accuracy for gender. It turned out that the misrecognized turns originate from the fact that the user performed off-talk with other persons in the background which caused the misrecognition. This finding suggests training the gender recognizer with non-speech and cross-talk samples in order to broaden the recognition from two (male, female) to three (male, female, non-speech) classes. Further it appears sensitive, to create a recognizer that would base its recognition on several speech samples instead of one, which would deliver a more robust result.

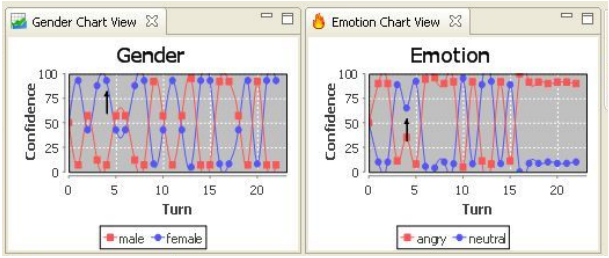

Figure 4: Screenshot of charts in Witchcraft based on turn-wise predictions an anger and a gender recognizer.

5 Portability towards other Corpora

Witchcraft has now been extended to cope with an unlimited number of corpora. An integration of new corpora is straight-forward. Witchcraft requires an SQL database containing two tables. The *dialogues* table hosts information on the overall dialogues (such as the dialogue ID, the category, filename of complete recording) and the *exchanges* table containing the turn-wise interactions (dialogue ID, turn number, system prompt, ASR parse, ASR confidence, semantic interpretation, hand transcription, utterance recording file, barged in, etc.). Both tables are linked through a $1 : n$ relationship, i.e. one entry in the dialogues table relates to n entries in the interactions table, cf. Figure 5. To demonstrate portability and in order to create a sample corpus that is deployed with Witchcraft, we included the CMU Let's Go bus information system from 2006 as demo corpus (Raux et al., 2006). It contains 328 dialogues including full recordings. The Witchcraft project includes a parser that allows to transform raw log data from the Let's Go system into the Witchcraft table structure.

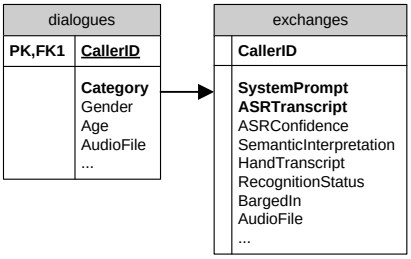

Figure 5: Dialogue and exchanges table with 1:n relationship. Bold database columns are required, others are optional.

6 Conclusion and Discussion

Witchcraft turned out to be a valuable framework in our everyday work when dealing with large dialogue corpora. At the current stage several students are working with it in multi-user mode to listen, analyze and annotate dialogues from three different corpora consisting of up to 100,000 dialogues each. Witchcraft allows them to search for dialogues relevant to the current task. The SQL-based access allows a powerful and standardized querying and retrieval of dialogues from the database. Witchcraft provides an overview and presents decisive information about the dialogue at one glance and allows to sort and group different types of dialogue for further research. Moreover, Witchcraft allows us to test arbitrary recognizers that provide additional information to the dialogue manager. Witchcraft tells us at which point in time a dialogue system would possess which knowledge. Further it allows us to conclude the reliability of this knowledge for further employment in the dialogue. For an evaluation of recognizers within Witchcraft please refer to (Schmitt et al., 2010b) where the deployment of an anger recognizer is simulated.

Witchcraft is now freely and publically available to the community. It is hosted under GNU General Public License at Sourceforge under witchcraftwb.sourceforge.org. 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. We hope that Witchcraft will help to foster research on future dialogue systems and we encourage the community to contribute.

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 CMU Let’s Go Lab from Carnegie Mellon University in Pittsburgh for their permission to deploy the Let’s Go Bus Information Corpus jointly with Witchcraft.

References

Niels Ole Bernsen, Laila Dybkjaer, and Mykola Kolodnytsky. 2002. The nite workbench - a tool for annotation of natural interactivity and multimodal data. In *Proc. of LREC*, pages 43–49, Las Palmas, Spain.

Tobias Bocklet, Andreas Maier, Josef Bauer, Felix Burkhardt, and Elmar Nöth. 2008. Age and gender recognition for telephone applications based on gmm supervectors and support vector machines. In *Proc. of ICASSP*, volume 1, pages 1605–1608.

Felix Burkhardt, Florian Metze, and Joachim Stegmann, 2007. *Speaker Classification for Next Generation Voice Dialog Systems*. Advances in Digital Speech Transmission. Wiley.

Klaus-Peter Engelbrecht, Florian Gödde, Felix Hartard, Hamed Ketabdar, and Sebastian Möller. 2009. Modeling user satisfaction with hidden markov model. In *Proc. of SIGDIAL 2009*, pages 170–177.

Antoine Raux, Dan Bohus, Brian Langner, Alan W. Black, and Maxine Eskenazi. 2006. Doing research on a deployed spoken dialogue system: One year of lets go! experience. In *Proc. of Interspeech*, September.

Alexander Schmitt, Gregor Bertrand, Tobias Heinroth, and Jackson Liscombe. 2010a. Witchcraft: A workbench for intelligent exploration of human computer conversations. In *Proc. of LREC*, Valetta, Malta, May.

Alexander Schmitt, Tim Polzehl, and Wolfgang Minker. 2010b. Facing reality: Simulating deployment of anger recognition in ivr systems. In *Proc. of IWSDS*, September.

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.