A Parameterized and Annotated Spoken Dialog Corpus of the CMU Let's Go Bus Information System

Alexander Schmitt, Stefan Ultes and Wolfgang Minker

Institute for Communications Engineering, University of Ulm, Germany
{alexander.schmitt, stefan.ultes, wolfgang.minker}@uni-ulm.de

Abstract

Standardized corpora are the foundation for spoken language research. In this work, we introduce an annotated and standardized corpus in the Spoken Dialog Systems (SDS) domain. Data from the Let's Go Bus Information System from the Carnegie Mellon University in Pittsburgh has been formatted, parameterized and annotated with quality, emotion, and task success labels containing 347 dialogs with 9,083 *system-user exchanges*. A total of 46 parameters have been derived automatically and semi-automatically from Automatic Speech Recognition (ASR), Spoken Language Understanding (SLU) and Dialog Manager (DM) properties. To each spoken user utterance an emotion label from the set *garbage, non-angry, slightly angry, very angry* has been assigned. In addition, a manual annotation of Interaction Quality (IQ) on the *exchange level* has been performed with three raters achieving a κ value of 0.54. The IQ score expresses the quality of the interaction up to each system-user exchange on a score from 1-5. The presented corpus is intended as a standardized basis for classification and evaluation tasks regarding task success prediction, dialog quality estimation or emotion recognition to foster comparability between different approaches on these fields.

Keywords: interaction quality, dialog performance measure, speech corpus

1. Introduction

Increasingly, data-driven techniques are employed in Spoken Dialog Systems (SDS) research with the aim of rendering SDSs more user-friendly and adaptive. As most studies rely on proprietary and non-publicly available corpora and as resources for comparisons are sparse, a transparent assessment of novel techniques is hardly possible. The Language Technology Institute (LTI) at Carnegie Mellon University (CMU) in Pittsburgh has taken the initiative to make results comparable within the framework of scientific and research activities by introducing the Spoken Dialog Challenge (Black and Eskenazi, 2009). For this purpose, both, the architecture and source code of the CMU Let's Go Bus Information system[1] as well as Let's Go interaction data collected in the field have been made publicly available. While we consider this as a cornerstone towards more transparency in research, we still felt the need for standard SDS corpora with a clearly defined scope and manageable size facilitating comparisons. Our aim in this contribution is to provide such a standardized, parameterized and well-defined corpus of manageable size that may be used in a variety of data-driven classification tasks. We propose to use this corpus for evaluating classifiers e.g., in assessing user satisfaction, and moreover, for acoustic and linguistic classification tasks, such as in emotion recognition as well as age and gender detection. The data set is based on Let's Go data from 2006 and has been preprocessed and parameterized with interaction parameters. Moreover, it has been manually annotated with interaction quality scores both, on dialog and exchange level, task completion labels and emotional states. The corpus is presented in such a manner that it may be promptly used for machine learning.

[1] Let's Go delivers bus schedule information to citizens of the city of Pittsburgh. It was created at Carnegie Mellon University (CMU) and answers 40-60 calls a day (in 2006), (Raux et al., 2006).

2. State of the Art

Pattern classification will allow adaptiveness in future HCI. This may be achieved by introducing statistical classifiers that use learned patterns to predict interaction-related knowledge. For example, (Walker et al., 2002) introduced a classifier estimating task success in spoken dialog. The industrial corpus that was applied for this study was derived from the AT&T How May I Help You (Gorin et al., 1996) system and is consequently not publicly available. In a similar spirit (Paek and Horvitz, 2004) and (Schmitt et al., 2010b) have implemented task success prediction. Evaluation has respectively been conducted on closed corpora from Microsoft and SpeechCycle, both not available to the community. Studies dealing with modeling acoustic properties of user speech, such as emotions, age or gender categories, further frequently employ closed industrial data sets, e.g., (Metze et al., 2007), (Schmitt et al., 2010a), (Lee and Narayanan, 2005). Also studies addressing automatic evaluation of SDS, such as PARADISE-style models (Walker et al., 2000) are based on closed data sets, see also (Engelbrecht et al., 2008), (Möller, 2005) and (Rieser and Lemon, 2008).

3. Corpus Preprocessing

Our contribution can be summarized as follows

- Formatting and Parameterization: Raw data from the log files has been transformed to a well-defined format, where each system-user exchange is represented as one logic entity. Each exchange has been parameterized with interaction parameters that quantify the interaction behavior of the user. The parameters may serve as input variables for a variety of classification tasks.
- Annotation: The corpus has been annotated with a number of labels that may serve as target variables

for classification. They are the Interaction Quality (Schmitt et al., 2011), i.e., expert quality scores for each exchange and the emotional state.

4. Formatting and Parameterization

The raw log information has been transferred to a common structure of a system-initiative directed spoken dialog, where an exchange e comprises interaction data from system turn s and user turn u, cf. Figure 1.

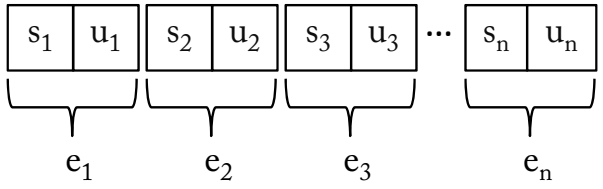

Figure 1: Structure of a system-initiative, directed spoken dialog.

Interaction parameters were created to model each exchange as well as its immediate context and further summarize the interaction that has taken place *until* the current exchange. For this, we use parameters on three levels: *exchange level* parameters, *window level* parameters, and *dialog level* parameters. The modeling levels are depicted in Figure 2.

On the *exchange level*, we modeled each system-user exchange with a number of Speech Recognition (ASR), Spoken Language Understanding (SLU) and Dialog Manager (DM)-related features. These features are automatically derived from system log files. A list of all used features can be found in Table 1.

To account for the overall history of important system events, we introduced *dialog level* parameters by adding running tallies, percentages and mean values for certain features symbolized with the suffixes '#', '%' and 'MEAN'.

Further, we consider the immediate context within the previous 3 turns of the current turn as particularly relevant for predicting target variables. Hence, derived from the basic parameters we created further *window level* parameters that emphasize specific user behavior prior to the classification point. They are symbolized with the prefix {#} for a number and {Mean} for the mean value.

We further introduced a semi-automatically determined dialog act feature group:

DAct SYSTEMDIALOGACT: one of 28 distinct dialog acts, such as *greeting, offer_help, ask_bus, confirm_departure, deliver_result, etc.* USERDIALOGACT: one of 22 distinct DAs, such as *confirm_departure, place_information, polite, reject_time, request_help*, etc.

All presented features are calculated automatically, with exception of DAct, without manual annotation or intervention, which would make them for a real-time deployment suitable.

5. Annotations (Target variables)

The corpus has been annotated with a number of target variables, which are

Table 2: Details of parameterized and annotated Let's Go corpus

# Dialogs	347
# System-User Exchanges	9,083
# Exchanges/Dialog	26.0 $\pm$ 21.5
Avg. Dialog Duration in seconds	116 s $\pm$ 114
Avg. User Turn Duration in seconds	1.5 s $\pm$ 1.9
# Dialogs with Emotion Labels	200
# Exchanges with Emotion Labels	4,885
# Raters	1
# Dialogs with IQ labels	200
# Exchanges with IQ labels	4,885
# Raters	3
Cohen's κ	0.54
Spearman's ρ	0.72

IQ For our work in (Schmitt et al., 2011), we annotated the corpus with Interaction Quality scores. Three raters annotated 200 dialogs (each dialog was rated by each rater) comprising 4,885 system-user exchanges. The raters were asked to annotate the quality of the interaction at each *system-user exchange* with the scores 5 (satisfied), 4 (slightly unsatisfied), 3 (unsatisfied), 2 (strongly unsatisfied) and 1 (extremely unsatisfied). To ensure quality, guidelines for the annotation have been developed beforehand. These can be seen in Table 3. Every dialog is initially rated with a score of 5 since in every interaction at the beginning the user can be considered as being satisfied with the dialog until the opposite eventuates. The final IQ score for an exchange is determined by creating the median of all three raters.

Emo We further introduce the negative emotional state of the user that is manually annotated by a human rater who chooses one of the labels *garbage, non-angry, slightly angry, very angry* for each single user turn. From all 4,832 user turns, 68.5% were non-angry, 14.3% slightly angry, 5.0% very angry and 12.2% contained garbage, i.e., non-speech events.

Task Success Each call has been annotated semi-automatically with a Task Success label, which is one of *completed* (187), *failed due to system behaviour* (15), *found out that there is no solution* (52), *not completed* (71) and *partially completed* (3)[2]. This was derived using a heuristic scheme, where the number of REQUEST, CONFIRMATION, and ERROR actions and the number of NO-MATCHES has been used. We will briefly describe the function of the heuristic. If at least one CONFIRMATION action has been performed, the call is labeled as *completed* in case of no left

[2]The number in brackets denote the label frequency.

Table 1: **Automatically derived features of the parameterized and annotated Let's Go corpus** On the *dialog level*, all features are calculated with respect to the whole dialog up to the current exchange. On the *window level*, only the last three exchanges were taken into account (cf. Fig. 2). Assignments of *dialog level* and *window level* features to either **ASR**, **SLU**, or **Dialog Manager** are equal to the according *exchange level* features.

	exchange level
ASR	
ASRRECOGNITIONSTATUS	one of 'success', 'reject', 'timeout'
ASRCONFIDENCE	confidence of the ASR
BARGED-IN?	did the user barge-in?
MODALITY	one of 'speech', 'DTMF'
EXMO	the modality expected from the system ('speech', 'DTMF', 'both')
UNEXMO?	did the user employ another modality than expected?
GRAMMARNAMES	names of the active grammars
TRIGGEREDGRAMMAR	name of grammar that matched
UTTERANCE	raw ASR transcription
WPUT	number of words per user turn
UTD	utterance turn duration
SLU	
SEMANTICPARSE	semantic interpretation of utterance
HELPREQUEST?	is the current turn a help request?
OPERATORREQUEST?	is the current turn an operator request?
Dialog Manager	
ACTIVITY	identifier of the current system action
ACTIVITYTYPE	one of 'question', 'announcement', 'wait_for_user_feedback'
PROMPT	system prompt
WPST	number of words per system turn
REPROMPT?	is the current system turn a reprompt?
CONFIRMATION?	whether the current system prompt is a confirmation to elicit common ground between user and system due to low ASR confidence
TURNNUMBER	current turn
DD	dialog duration up to this point in seconds
	dialog level
MEANASRCONFIDENCE	average of ASR confidence scores
#ASRSUCCESS	number of exchanges with ASRRECOGNITIONSTATUS 'success'
%ASRSUCCESS	rate of exchanges with ASRRECOGNITIONSTATUS 'success'
#ASRREJECTIONS	number of exchanges with ASRRECOGNITIONSTATUS 'reject'
%ASRREJECTIONS	rate of exchanges with ASRRECOGNITIONSTATUS 'reject'
#TIME-OUTPROMPTS	number of exchanges with ASRRECOGNITIONSTATUS 'timeout'
%TIME-OUTPROMPTS	rate of exchanges with ASRRECOGNITIONSTATUS 'timeout'
#BARGEINS	number of barge-ins
%BARGEINS	rate of barge-ins
#UNEXMO	number of turns with unexpected modality
%UNEXMO	rate of turns with unexpected modality
	window level
{MEAN}ASRCONFIDENCE	average of ASR confidence scores
{#}ASRSUCCESS	number of successfully parsed user utterances
{#}ASRREJECTIONS	number of exchanges with ASRRECOGNITIONSTATUS 'reject'
{#}TIME-OUTPROMPTS	number of exchanges with ASRRECOGNITIONSTATUS 'timeout'
{#}BARGEINS	number of barge-ins
{#}UNEXMO	number of turns with unexpected modality
{#}HELPREQUESTS	number of turns where user requested help
{#}OPERATORREQUESTS	number of turns where user requested an operator
{#}REPROMT	number of turns with reprompt
{#}CONFIRMATION	number of turns where the system prompt is a confirmation
{#}SYSTEMQUESTIONS	number of turns where ACTIVITYTYPE is 'question'

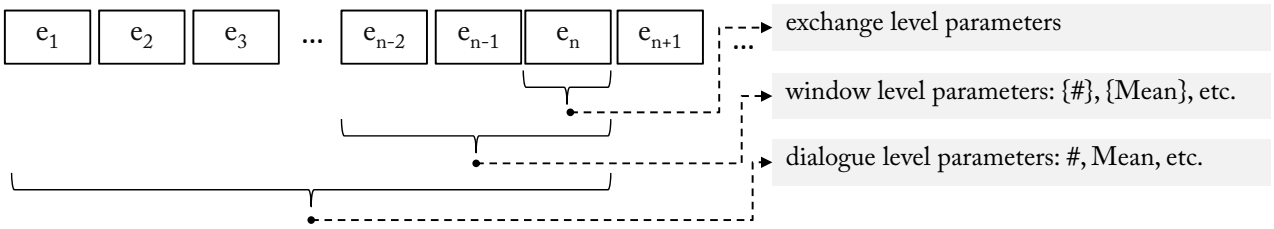

Figure 2: The three different modeling levels representing the interaction at exchange e_n: The most detailed exchange level, comprising parameters of the current exchange; the window level, capturing important parameters from the previous n dialog steps (here $n = 3$); the dialog level, measuring overall performance values from the entire previous interaction.

open requests, and as *partially completed* otherwise[3]. If no CONFIRMATION action has been performed at all, there are three possible distinctions: First, NO-MATCHES occurred. Then, the call was labeled as *failed due to system behaviour*. Second, an ERROR action has been performed (e.g., destination was not covered by the system). Then, the call was labeled as *found out that there is no solution*. Finally, the call was labeled as *not completed* for all other cases.

6. Corpus Details

The details of the corpus are depicted in Table 2. In order to take into account the ordinal character of the IQ scores, Cohen's κ has been used with additional weights w. These weights were determined by the numerical distance d between the ratings, resulting in

$$w = \frac{|d|}{k-1},$$

where k is the number of different IQ scores. By this, the penalty for small differences between the raters is not as high as for unweighted κ, which would be 1 for every mismatch. Achieving a κ value of 0.54 is a good result considering the difficulty of this task.

7. Download

The corpus is deployed in form of CSV files and SQL dumps and may be downloaded at `http://nt.uni-ulm.de/ds-lego`. We encourage to add additional labels and annotations to this data set.

8. Acknowledgements

We would like to thank Maxine Eskenazi, Alan Black, Lori Levin, Rita Singh, Antoine Raux and Brian Langner from the Let's Go Lab at Carnegie Mellon University, Pittsburgh, for providing the Let's Go Sample Corpus.

9. References

Alan Black and Maxine Eskenazi. 2009. The spoken dialogue challenge. In *Proceedings of the SIGDIAL 2009 Conference*, pages 337–340, London, UK, September. Association for Computational Linguistics.

Klaus-Peter Engelbrecht, Christine Kühnel, and Sebastian Möller. 2008. Weighting the coefficients in paradise models to increase their generalizability. In *Proceedings of the 4th IEEE tutorial and research workshop on Perception and Interactive Technologies for Speech-Based Systems: Perception in Multimodal Dialogue Systems*, PIT '08, pages 289–292, Berlin, Heidelberg. Springer-Verlag.

A. L. Gorin, B. A. Parker, R. M. Sachs, and J. G. Wilpon. 1996. How may i help you? In *Interactive Voice Technology for Telecommunications Applications, 1996. Proceedings., Third IEEE Workshop on*, pages 57–60, Sep-1 Oct.

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

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.

Sebastian Möller. 2005. *Quality of Telephone-based Spoken Dialogue Systems*. Springer, New York.

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

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 the International Conference on Speech and Language Processing (ICSLP)*, September.

Verena Rieser and Oliver Lemon. 2008. Automatic learning and evaluation of user-centered objective functions for dialogue system optimisation. In Bente Maegaard Nicoletta Calzolari (Conference Chair), Khalid Choukri, editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

Alexander Schmitt, Tim Polzehl, and Jackson Liscombe. 2010a. The influence of the utterance length on the

[3] A CONFIRMATION action was considered to fulfill all previous requests.

Table 3: Rater guidelines for annotating Interaction Quality

1.	The rater should try to mirror the user's point of view on the interaction as objectively as possible.
2.	An exchange consists of the system prompt and the user response. Due to system design, the latter is not always present.
3.	The IQ score is defined on a five-point scale with "1=extremely unsatisfied', "2=very unsatisfied", "3=unsatisfied", "4=slightly unsatisfied" and "5=satisfied".
4.	The Interaction Quality is to be rated for each exchange in the dialog. The history of the dialog should be kept in mind when assigning the score. For example, a dialog that has proceeded fairly poor for a long time, should require some time to recover.
5.	A dialog always starts with an Interaction Quality score of "5".
6.	The first user input should also be rated with 5, since until this moment, no rateable interaction has taken place.
7.	A request for help does not invariably cause a lower Interaction Quality, but can result in it.
8.	In general, the score from one exchange to the following exchange is increased or decreased by one point at the most.
9.	Exceptions, where the score can be decreased by two points are, e.g., hot anger or sudden frustration. The rater's perception is decisive here.
10.	Also, if the dialog obviously collapses due to system or user behavior, the score can be set to "1" immediately. An example therefore is a reasonable frustrated sudden hang-up.
11.	Anger does not need to influence the score, but can. The rater should try to figure out whether anger was caused by the dialog behavior or not.
12.	In the case a user realizes that he should adapt his dialog strategy to obtain the desired result or information and succeeded that way, the Interaction Quality score can be raised up to two points per turn. In other words, the user realizes that he caused the poor Interaction Quality by himself.
13.	If the system does not reply with a bus schedule to a specific user query and prompts that the request is out of scope, this can nevertheless be considered as "task completed". Therefore this does not need to affect the Interaction Quality.
14.	If a dialog consists of several independent queries, each query's quality is to be rated independently. The former dialog history should not be considered when a new query begins. However, the score provided for the first exchange should be equal to the last label of the previous query.
15.	If a constantly low-quality dialog finishes with a reasonable result, the Interaction Quality may be increased.

recognition of aged voices. In *International Conference on Language Resources and Evaluation (LREC)*, Valetta, Malta, May.

Alexander Schmitt, Michael Scholz, Wolfgang Minker, Jackson Liscombe, and David Sündermann. 2010b. Is it possible to predict task completion in automated troubleshooters? In *Proc. of the International Conference on Speech and Language Processing (ICSLP)*, September.

Alexander Schmitt, Benjamin Schatz, and Wolfgang Minker. 2011. Modeling and predicting quality in spoken human-computer interaction. In *Proceedings of the SIGDIAL 2011 Conference*, Portland, Oregon, USA. Association for Computational Linguistics.

Marilyn Walker, Candace Kamm, and Diane Litman. 2000. Towards developing general models of usability with paradise. *Nat. Lang. Eng.*, 6(3-4):363–377.

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.