ISCA Archive
http://www.isca-speech.org/archive

INTERSPEECH 2012
ISCA's 13th Annual Conference
Portland, OR, USA
September 9-13, 2012

Assessment of user simulators for spoken dialogue systems by means of subspace multidimensional clustering

Zoraida Callejas[1], *David Griol*[2], *Klaus-Peter Engelbrecht*[3]

[1]Department of Languages and Computer Systems, University of Granada, Granada, Spain
[2]Department of Computer Science, University Carlos III of Madrid, Leganés, Spain
[3]Quality and Usability Lab, Deutsche Telekom Laboratories, TU Berlin, Berlin, Germany
zoraida@ugr.es, dgriol@inf.uc3m.es, klaus-peter.engelbrecht@telekom.de

10.21437/Interspeech.2012-79

Abstract

The assessment of user simulators in terms of their similarity with real users implies processing and interpreting large dialogue corpora, for which many interaction parameters can be considered. In this setting, the high dimensionality of the data makes it difficult to compare the dialogues as it is not always appropriate to consider all features equally in order to carry out meaningful interpretations. We propose to use subspace clustering for the assessment of users simulators, as this technique has been successfully applied to tackle and classify high-dimensional information in other areas of study. We created and assessed a user simulator for the Let's Go spoken dialogue system. The experimental results show that the proposed approach is easy to set up and helps to better interpret whether the user simulator has similar behaviours to real human users by creating clusters with different dimensions which cannot be identified with plain clustering techniques.

Index Terms: user simulation, spoken dialogue systems, evaluation, clustering.

1. Introduction and background

In the recent years statistical methodologies have been successfully applied in several aspects of spoken dialogue systems development, such as dialogue management, speech recognition and understanding, natural language generation and evaluation of system performance [1, 2]. These techniques require large amounts of training data in order to explore wide search spaces and user simulators have appeared as a relevant opportunity to generate such data reducing the costs in terms of time and effort.

In this context, the assessment of the simulated user is necessary in order to choose the optimal model for simulation, and compute the reliability of the simulated behaviours. However, there are no generally accepted criteria to measure the quality of user simulators [3].

Since user simulations are built to replace human users, one intuition is that a good user simulation should be able to replicate human user behaviours. Under this assumption, assessing the user simulator usually consists in assessing the realism of the simulated used (e.g. with human judges in a kind of Turing test) or measuring the similarity to real user behaviour based on more formal criteria.

For example, some methods view the user model as a predictor of user actions and evaluate it by measuring the quality of its predictions with *recall* and *precision* [4]. *Recall* measures how many of the actions in the real response are predicted correctly, and *precision* measures the proportion of correct actions among all the predicted actions. Other approaches consist in comparing the generated dialogues in terms of statistics of interaction parameters [3, 5], which can be grouped into high-level features (e.g. dialogue and turn lengths) and dialogue style features (e.g. speech-act frequency or proportion of goal-directed actions). Such parameters may also be combined into a single performance measure and compared on this level. Next to graphical comparisons, metric distances have been calculated in several ways (e.g. [6]).

While a single distance measure is attractive for showing progress over a previous model, it is not very informative with respect to possible improvements of the model. On the other hand, a big set of features makes it difficult to comprehend and visualize the distance, loosing precision as the number of dimensions grows [7]. On the other hand, given the high number of features some of them may not be meaningful for the comparison [8]. To encounter this, we propose to carry out the assessment of the simulated dialogues using a technique called subspace clustering [9], which overcomes the negative effects of high dimensionality considering several subspaces in which clusters are built using a different number of dimensions. In order to evaluate our proposal we employed 338 real dialogues from the Let's Go system [10] and implemented a statistical user simulator with which 1,000 dialogues were generated (Section 3). Then, we computed a large set of dialogue parameters (Section 4) for the 1,338 dialogues, which were then automatically clustered. The results are discussed in Section 5, while conclusions and future work are presented in Section 6.

2. Clustering method

Clustering consists in grouping data into categories (clusters) so that the data in the same cluster are more similar to each other than the data in different clusters. Thus, if real and simulated dialogues are points in a clustering algorithm, the quality of the simulator can be assessed by studying whether they are clustered apart, and whether all types of dialogues are covered by the simulation.

In traditional clustering algorithms, similarity is computed according to distance measures calculated over a series of dimensions, in our case, dialogue features. These algorithms make the basic assumption of independence among the attributes, thus in many cases the data are pre-processed to make the attributes more independent. However, distance functions that equally use all input features may not be effective as it might be the case that some features might be important to find correlations between certain dialogues but not with others [7].

In order to solve this problem and eliminate the negative effects of the high dimensionality, we propose to use subspace clustering, which considers several subspaces with a different number of dimensions. This way each feature might be relevant to at least one of the clusters, and at the same time it is not necessary to reduce the dimensionality of the space beforehand, which may lead to a loss of valuable information.

For our experiments we have employed the PROCLUS projected clustering algorithm, which detects all the possible clusters in all subspaces. The algorithm builds the clusters taking into account different subsets of the attributes and assigns each point (in our case each dialogue) to a unique cluster. To do so, we used Opensubspace [11], an implementation of subspace algorithms that can be integrated into the Weka tool.

3. User simulator for Let's Go

To evaluate our proposal, we have developed a statistical user simulator for the Let's Go system. Let's Go is a spoken dialogue system that provides bus schedule information in Pittsburgh. It was made available for the general public in 2005 [12], which allowed to gather a large amount of data from spoken interactions of real users with the system. In 2009, the corpus was distributed among the scientific community as a common database for the 2010 Spoken Dialogue Challenge initiative [10].

The methodology that we have developed for the simulation of users' actions on the intention level extends our work for developing a statistical methodology for dialogue management [13]. The user answers are generated taking into account the information provided by the simulator throughout the history of the dialogue, the last system turn, and the objective(s) predefined for the dialogue. A labelled corpus of dialogues is used to estimate the user model using a MLP.

We represent the dialogues as a sequence of pairs (A_i, U_i), where A_i is the output of the system at time i, expressed in terms of dialogue acts; and U_i is the semantic representation of the user turn (the result of the understanding process of the user input) at time i, expressed in terms of frames: $(A_1, U_1), \cdots, (A_n, U_n)$. We refer to a pair (A_i, U_i) as S_i, the state of the dialogue sequence at time i.

As the number of all possible sequences of states is very large, we establish a partition with a data structure called *User Register* (UR), which contains the information about concepts and attribute values provided by the user throughout the previous history of the dialogue. Given this representation, the objective of the user simulator at time i is to find an appropriate user answer U_i. This selection, which is a local process for each time i, takes into account the sequence of dialogue states that precede time i, the system answer at time i, and the objective of the dialogue $\mathcal{O}$, which we compute as the maximization: $\hat{U}_i = _{U_i \in \mathcal{U}} P(U_i | UR_{i-1}, A_i, \mathcal{O})$.

For the Let's Go task, the variable $\mathcal{O}$ was modelled taking into account the different objectives labelled in the corpus of the Spoken Dialog Challenge by considering the different places and times for which the real users required information (from one to five), users' requirements about previous and next buses, number of uncovered places, and possible system failures. The different combinations of these parameters in the corpus led to the definition of 38 different objectives.

Additionally, an error simulator module has been implemented to perform error generation. The error simulator modifies the frames generated by the user simulator once the UR is updated. In addition, the error simulator adds a confidence score to each concept and attribute in the frames. Both processes are carried out separately following the noisy communication channel metaphor by means of a generative probabilistic model as in [14].

4. Dialogue features

In total, 21 features have been computed. Firstly, we computed the following interaction parameters:

- Number of exchanges between user and system ($numExchanges$). An exchange is comprised of a system turn and the successive user turn, where the user turn can be empty (e.g. no-input or when the end of a dialogue is reached).
- Percentage of confirmation dialogue acts ($percConfirm$), mean and maximum re-asks ($meanReasks$ and $maxReasks$).
- Number and percentage of no-matches ($numNoMatches$, $percNoMatches$) and no-inputs ($numNoInputs$, $percNoInputs$).

- Mean number of concepts provided in the user utterances ($meanAVPs$). A concept can be a value for a slot, a logical value (yes/no), a dtmf signal, navigation keywords such as next bus, help requests or a dialogue ending action. The total number of concepts is divided by the number of exchanges in the dialogue to obtain the mean.

- Task success ($successDial$), determined automatically by checking whether the objective of the dialogue was reached.

- Number of system dialogue acts in a dialogue divided by the total number of concepts provided by the user ($SysActPerUserAct$).

Secondly, the system dialogue acts were classified into 5 groups: *formal* (dialogue formalities like "welcome"), *results* (presentation of search results), *queries* (request for values to fill slots), *statusReports* (when the system provides feedback about its status, e.g. "looking up database"), *error* (error messages) and *instructions* (instructions to tell the user how to speak to the system). The counts and percentages of each group were calculated as new parameters.

5. Discussion of results

Table 1 shows the 4 clusters generated. As can be observed, different features have been chosen for each cluster, and thus there are four-dimensional and two-dimensional subspaces. The features selected are mainly related to situations in the dialogue which differ from the optimal, such as out-of-vocabulary inputs, silences, number of error messages, percentage of error messages and the percentage of dialogue acts corresponding to presentation of results. There were 150 unclassified dialogues, from which 134 were real and 16 simulated.

Cluster (dimensions): *relevant features*	#Dialogues
0 (4D): *numNoInputs*, *percNoInputs*, *successDial*, *percResults*	59
1 (4D): *percNoInputs*, *meanAVP*, *numError*, *percError*	861
2 (2D): *numNoMatches*, *numNoInputs*	239
3 (2D): *percNoInputs*, *percResults*	29
Unclustered:	150

Table 1: Results of the subspace clustering for the Let's Go task

We have carried out a statistical study per cluster computing the maximum, minimum, average and standard deviation of the parameters (Table 2 shows average values). The study reveals that the dialogues in cluster 1 mostly reached their objectives and there were no error dialogue acts. Cluster 2 is comprised of dialogues in which there were no no-matches and no-inputs, and however, the percentage of error dialogue acts was the highest and the task success the lowest. This might indicate that the low dialogue success could be due to uncovered requests, as the users' inputs were correctly recognized (no no-matches). Cluster 0 and cluster 3, which hold a reduced amount of the dialogues, are comprised of the longest interactions. The case of cluster 3 is peculiar, as it holds the dialogues with the higher number of no-matches and no-inputs.

Parameters	Cluster 0	Cluster 1	Cluster 2	Cluster 3
numExchanges	18.10	9.76	9.69	30.14
percConfirm	0.62	0.59	0.57	0.66
numNoMatches	3.61	0.40	0.00	10.52
numNoInputs	0.00	0.00	0.00	0.04
meanAVPs	1.03	1.00	0.99	1.04
successDial	1.00	0.97	0.46	1.00
sysActPerUserAct	1.76	2.12	2.33	1.53
meanReasks	1.28	1.09	1.15	1.58
maxReasks	2.78	1.56	1.69	4.24
percNoMatches	0.18	0.03	0.00	0.03
percNoInputs	0.00	0.00	0.00	0.01
Dialogue acts	Cluster 0	Cluster 1	Cluster 2	Cluster 3
numFormal	1.19	1.12	1.08	1.03
numResults	1.81	1.43	1.13	1.59
numQueries	13.78	7.99	6.97	22.41
numStatusReports	8.39	5.15	5.22	10.73
numError	1.02	0.00	1.06	3.24
numInstructions	3.39	2.31	2.09	4.17
percFormal	0.08	0.13	0.19	0.05
percResults	0.10	0.15	0.07	0.05
percQueries	0.78	0.82	0.62	0.76
percStatusReports	0.46	0.53	0.42	0.04
percError	0.06	0.00	0.26	0.10
percInstructions	0.20	0.25	0.27	0.15

Table 2: Average value of the interaction parameters in each cluster

Additionally, clustering can be used to group dialogues with many different objectives to a few groups, allowing to analyse the data cluster-wise. The users of the Let's Go system follow different objectives, e.g. some users simply look for a line connecting 2 stops, others need departure times, and others need information about complete connections. Completing these tasks with the system can involve navigation within the results, and in some cases also changing the query, e.g. if two stops are not connected by a line. Cluster 1 mainly comprises short dialogues, and cluster 2 contains normal searches for a connection with 2 stops. Dialogues with more than 2 stops are distributed across all clusters except cluster 1.

The method also allows comparing the datasets graphically. Figure 1 shows how the datasets are differently structured with regard of the distribution of the clusters. Figure 1 (bottom) shows that most dialogues of both datasets are in clusters 1 and 2. As those clusters contain more standard interactions, it seems that the simulator was able to render a realistic behaviour. However, as shown in Figure 1 (top), clusters 1 and 2 are mainly built based on the simulated data, whereas the real data contributed mostly to clusters 0 and 3. This shows that

the features used to define the subspaces for those clusters might indicate a difference in the behaviour of the simulated and real dialogues, concretely the number of no inputs seems to be relevant. The fact that there are no simulated dialogues in cluster 3, highlights the difficulty of developing a realistic mechanism to render no inputs automatically.

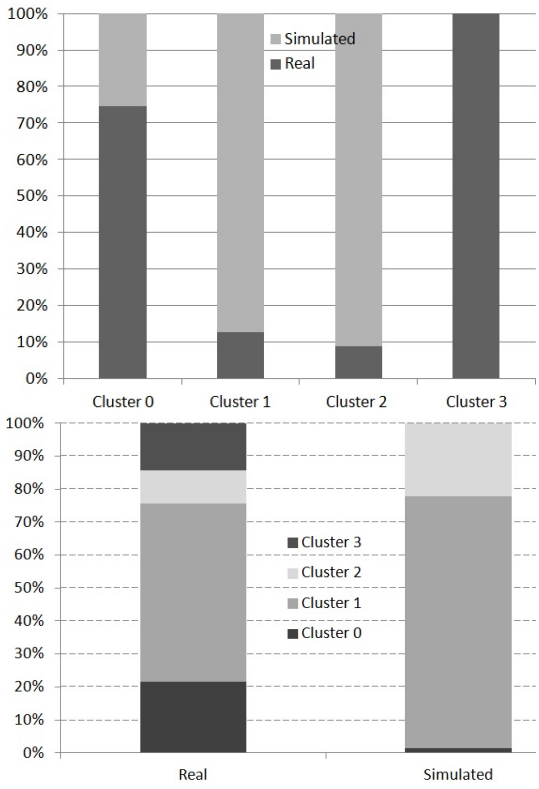

Figure 1: Distribution of real and simulated dialogues among clusters

6. Conclusions and future work

In the paper we have described how to use subspace clustering to assess the quality of user simulators in spoken dialogue systems. It is based on clustering real and simulated users in a space whose dimensions are the interaction parameters. To evaluate the proposal, we have developed and assessed a statistical user simulator for the Let's Go spoken dialogue system. The results show that different features might be relevant to build different clusters and the analysis of such clusters allows to easily assess the quality of the simulator and discover the differences between the simulated and real users in order to improve the simulation model. For future work we plan to apply the proposed technique to other tasks and also to other simulation models for the same task in order to see whether it is also applicable for comparison between several simulated users.

7. References

[1] K.-P. Engelbrecht, M. Quadeb, and S. Moeller, "Analysis of a new simulation approach to dialog system evaluation," *Speech Communication*, vol. 51, no. 12, pp. 1234 – 1252, 2009.

[2] O. Lemon, "Learning what to say and how to say it: Joint optimisation of spoken dialogue management and natural language generation," *Computer Speech and Language*, vol. 25, no. 2, pp. 210 – 221, 2011.

[3] O. Pietquin and H. Hastie, "A survey on metrics for the evaluation of user simulations," *Knowledge Engineering Review*, vol. In Press, 2011.

[4] J. Schatzmann, K. Georgila, and S. Young, "Quantitative Evaluation of User Simulation Techniques for Spoken Dialogue Systems," in *SIGdial*, 2005, pp. 45–54.

[5] D. Griol, Z. Callejas, and R. López-Cózar, "A comparison between dialog corpora acquired with real and simulated users," in *SIGdial*, 2009.

[6] J. D. Williams, "Evaluating user simulations with the cramér von mises divergence," *Speech Communication*, vol. 50, no. 10, pp. 829 – 846, 2008.

[7] W. Wang and J. Yang, "Mining high-dimensional data," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Springer, 2010, pp. 803–808.

[8] Z. Callejas and R. López-Cózar, "Relations between de-facto criteria in the evaluation of a spoken dialogue system," *Speech Communication*, vol. 50, no. 8-9, pp. 646–665, 2008.

[9] R. Vidal, "Subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52 – 68, 2011.

[10] A. Black, S. Burger, B. Langner, G. Parent, and M. Eskenazi, "Spoken dialog challenge 2010," in *IEEE SLT*, 2010.

[11] E. Muller, S. Gunnemann, I. Assent, and T. Seidl, "Evaluating clustering in subspace projections of high dimensional data," in *VLDB*, 2009.

[12] A. Raux, B. Langner, A. Black, and M. Eskenazi, "Let's go public! taking a spoken dialog system to the real world," in *Interspeech*, 2005.

[13] D. Griol, L. Hurtado, E. Segarra, and E. Sanchis, "A Statistical Approach to Spoken Dialog Systems Design and Evaluation," *Speech Communication*, vol. 50, no. 8–9, pp. 666–682, 2008.

[14] J. Schatzmann, B. Thomson, and S. Young., "Error Simulation for Training Statistical Dialogue Systems," in *ASRU*, 2007, pp. 526–531.