Correlation-based Query Relaxation for Example-based Dialog Modeling

Cheongjae Lee, Sungjin Lee, Sangkeun Jung, Kyungduk Kim, Donghyeon Lee, Gary Geunbae Lee

Department of Computer Science and Engineering

Pohang University of Science and Technology (POSTECH), South Korea

{lcj80,junion,hugman,getta,semko,gblee}@postech.ac.kr

***Abstract*—Query relaxation refers to the process of reducing the number of constraints on a query if it returns no result when searching a database. This is an important process to enable extraction of an appropriate number of query results because queries that are too strictly constrained may return no result, whereas queries that are too loosely constrained may return too many results. This paper proposes an automated method of correlation-based query relaxation (CBQR) to select an appropriate constraint subset. The example-based dialog modeling framework was used to validate our algorithm. Preliminary results show that the proposed method facilitates the automation of query relaxation. We believe that the CBQR algorithm effectively relaxes constraints on failed queries to return more dialog examples.**

I. Introduction

A query consists of a set of attribute-value pairs in a relational database. These pairs are carefully chosen to include only a specified subset of the database. If constraints on the query are too strict, it may fail to produce an answer. Constraints on queries that fail to produce an answer (failed queries) are relaxed to satisfy more tuples so that queries can return relevant answers [1-4].

We here suggest a simple correlation-based algorithm for relaxing constraints on failed queries for an example-based dialog modeling (EBDM) framework [5]. In the EBDM framework, a query (e.g., a set of semantic and discourse constraints) returns semantically-similar dialog examples to predict the next system action. However, failed queries are inevitable because of the data sparseness problem or because of prior errors by the speech recognition and language understanding modules. To avoid this problem, query constraints are relaxed using heuristic relaxation strategies so that more matches are found. However, these strategies cannot cover all situations perfectly because the strategies are prepared manually by human experts and it is difficult to know which query constraints should be relaxed in a specific situation. Therefore, this paper proposes a correlation-based query relaxation (CBQR) algorithm that uses correlation coefficients between pairs of query constraints to automatically select a constraint subset by removing irrelevant constraints.

II. Related Work

Some approaches to query relaxation have been proposed which relax constraints on failed queries by selecting a subset of query constraints. For example, a query relaxation method using domain-independent strategies was used to relax failed queries to find more contents (e.g., restaurants and MP3 songs) for information presentation in a spoken dialog system [1]. In addition, automated query relaxation algorithm was proposed that uses domain knowledge extracted from a small dataset to generate successful queries that are highly similar to the failed query [2]. Previous work on query relaxation for spoken dialog systems addressed the problem of relaxing the user's constraints on the database query to help the user find an appropriate item. This paper, however, applies this technique to retrieve dialog examples in a dialog example database because manually relaxing failed queries is laborious and time-consuming.

Our approach also addresses the problem of state generalization for dialog models [6, 7]. For example, detailed dialog states in a master space are grouped into general dialog states in a summary space to apply partially observable Markov decision process (POMDP) framework. Although the dialog states in the summary space can be automatically obtained with a clustering technique [8], we can automatically generalize the dialog states based on the correlation relationship between its state variables (or query constraints) to find the similar dialog examples.

To remove irrelevant query constraints, our query relaxation algorithm is inspired by correlation-based feature selection algorithms [9, 10]. In the classification task, an appropriate subset of features can be selected by removing irrelevant and redundant features because a feature is useful if it is correlated with a target class. While feature-class correlations (e.g., correlation coefficient, mutual information, and chi-square statics) are considered to minimize a classification error rate, our proposed method considers only correlations between pairs of query constraints because we assume that a relevant constraint subset is one that contains constraints that are positively correlated with each other.

TABLE I
LIST OF INDEX AND QUERY CONSTRAINTS USED IN THE EBDM FRAMEWORK

Constraints	Detail Descriptions
Dialog Act (DA)	Domain-independent label of an utterance at the level of illocutionary force (e.g. STATEMENT, REQUEST, WH-QUESTION)
Main Goal (MG)	Domain-dependent user goal of an utterance (e.g. GUIDE-LOC, SEARCH-LOC, SEARCH-PER-MAIL)
Slot Flag (SF)	Filling flag of the corresponding slot name at the current turn (e.g. If user says "meeting room" at the current turn, the flag of LOC_ROOM_NAME is set to 1.)
Previous Dialog Act (PrevDA)	Dialog act of the previous user utterance
Previous Main Goal (PrevMG)	Main goal of the previous user utterance
Discourse History Vector (DHV)	Binary vector for slot-filling status that is assigned a value of 1 if the component slot is already filled, and 0 otherwise.

In this paper, we are interested in exploring an automated query relaxation algorithm for EBDM. We investigated whether the CBQR appropriately relaxes constraints on failed queries when the dialog manager finds no dialog example in the EBDM framework.

III. QUERY RELAXATION PROBLEM

EBDM is a data-driven dialog modeling technique in which the next system action is determined by selecting the most similar dialog example in a dialog example database (DEDB). To generalize the data, a dialog example is first indexed as a set of tuples that have the same semantic and discourse constraints (Table I). A set of the semantic and discourse constraints is similar to a set of state variables to represents the dialog state. Details of how EBDM framework determines the next system action are summarized in Lee et al. [5].

The example search step in the EBDM framework is similar to the state generalization problem because the dialog state space is too huge to cover all situations with limited examples. Thus, if the dialog manager returns no examples, the constraints should be relaxed to generalize the current dialog state. For example, if the query for the exact match

$$Q_f:\ DA='a' \wedge MG='b' \wedge PrevDA='c' \wedge PrevMG='d'$$

fails because the constraints may be too strict or if some constraints contain errors, relaxation strategies are used to relax particular query constraints so that more examples can be found. Then, the following relaxed query is generated heuristically:

$$Q_r:DA='a' \wedge MG='b'$$

because the system action is more dependent on the current input than on the previous input.

Constraints on the failed query are relaxed by using heuristic relaxation strategies according to the importance and reliability of each constraint, which are predefined by human experts. The aim of each relation strategy is to satisfy more dialog examples by removing less-reliable and less-important constraints on the failed query. However, defining the heuristic relaxation strategies has the serious problem that estimating the reliability and importance of each constraint is very difficult. In fact, the best performance can be achieved by selecting the best example among exactly-matched examples in the DEDB at every turn. However, over-constrained queries (e.g., exact match) often return no examples because the dialog state space is extremely large but the number of dialog examples is limited in the dialog example database. In addition, like over-relaxing the query, it may require prohibitive computational time to select the best example among returned examples because too many candidate examples exist that satisfy the query. Therefore, selecting a suitable subset of query constraints is important for both computational efficiency and dialog performance.

To address these problems, we sought an automated CBQR algorithm which can search for semantically-similar dialog examples in the EBDM framework.

IV. CORRELATION-BASED QUERY RELAXATION

For a set of k query constraints $Q = \{q_i | i = 1, \dots, k\}$, the linear correlation coefficient r_{ij} between constraints q_i and q_j for a query constraint set in correlation matrix R is defined as:

$$r_{ij} = corr(q_i, q_j) = \frac{\mathrm{cov}(q_i, q_j)}{\sigma_{q_i}\sigma_{q_j}}$$

where $\mathrm{cov}(\cdot)$ is the covariance, and σ is the standard deviation of corresponding query constraint. The value of r_{ij} ranges from -1 (perfectly negatively correlated) to 1 (perfectly positively correlated). If $r_{ij} = 1$, then when constraint q_i occurs, constraint q_j also always occurs. If $r_{ij} = -1$, then when constraint q_i occurs, constraint q_j never occurs, and vice versa. Instances in which constraint q_i occurs and constraint q_j may or may not occur (and vice versa) yield $-1 \le r_{ij} \le 1$. The proposed algorithm exploits this relationship by selecting only pairs of constraints for which r_{ij} exceeds some threshold.

The goal of the CBQR algorithm is to find a new constraint subset $\hat{Q}$ in which irrelevant constraints are eliminated based on their correlation coefficients in the original set Q (Fig. 1).

Note that the most correlated query constraint is the constraint q^* that has the greatest sum of correlation

Algorithm : CBQR(Q,R)

Input : $Q = \{q_i \mid i = 1,\ldots,k\}$: original query constraint set
$R = \{r_{ij} \equiv corr(q_i, q_j) \mid i, j = 1,\ldots,k\}$: correlation matrix

Output : $\hat{Q} = \{q_i \mid i = 1,\ldots,m; m < k\}$: relaxed query constraint subset

1 $q^* \leftarrow \arg\max_{q_i \in Q} \sum_{j=1}^{k} corr(q_i, q_j)$
2 $\hat{Q} \leftarrow Q$
3 for $i \leftarrow 1$ to k
4 do $\hat{Q} \leftarrow \hat{Q} - \{q_i \mid corr(q_i, q^*) \le \theta\}$
5 return $\hat{Q}$

Fig. 1. CBQR algorithm

coefficients between itself and other constraints. q^* is not optimal but good to be representative of the constraints because q^* may co-occur frequently with other constraints. This is similar to correlation-based feature selection method in which more correlated features to certain class are critical to classify it. Consequently, q^* is calculated as:

$$q^* = \arg\max_{q_i \in Q} \sum_{j=1}^{k} corr(q_i, q_j)$$

Any constraints that is not highly correlated with q^* are eliminated from the original set Q when the correlation coefficient between that query constraint and q^* falls below a preset threshold θ. The eliminated constraints may be negatively correlated with, or independent of q^*, which indicates that they seldom co-occur with q^* in the same dialog state. The irrelevant constraints may be the erroneous constraints generated by prior modules. Therefore, we believe that our approach can be robust to prior errors because the erroneous constraints may be eliminated automatically. θ is empirically assigned according to the correlation distribution. It can be set higher to consider only strongly positively-correlated constraints.

Finally, the CBQR creates a relaxed constraint subset $\hat{Q}$ that contains only constraints that are positively correlated with q^*. Note that $\hat{Q}$ satisfies at least the dialog examples covered by Q. This implies that $\hat{Q}$ is guaranteed not to exclude dialog examples returned by Q.

At the example search step, $\hat{Q}$ is first used to find semantically-similar examples. However, if $\hat{Q}$ returns no example, $\hat{Q}$ could be exhaustively relaxed by removing the most irrelevant constraint as follows:

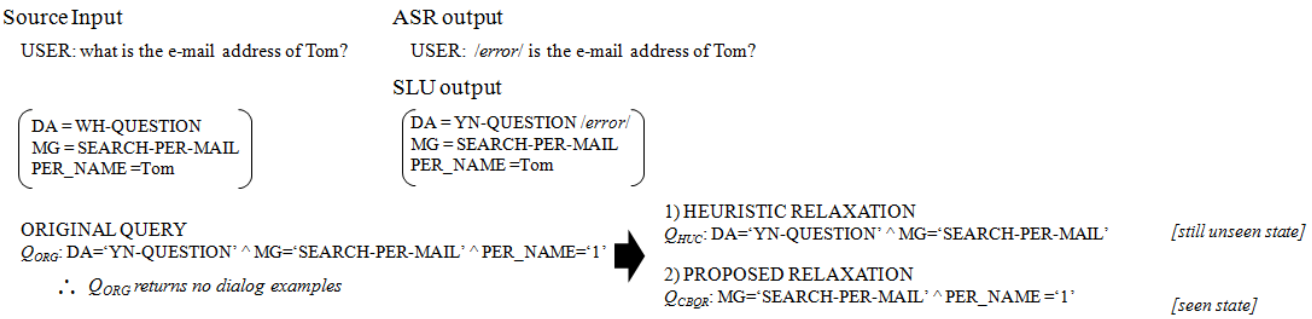
$$\hat{Q} \leftarrow \hat{Q} - \{q_i \mid \text{argmin}_{q_i \in \hat{Q}}\, corr(q^*, q_i)\}$$

Fig. 2 illustrates a concrete example of our proposed approach. The user says "*what is the e-mail address of Tom*", but some errors occur due to wrong predictions in ASR and SLU modules. In this case, Q_{ORG} returns no dialog examples because the co-occurrence of 'DA=YN-QUESTION' and 'MG=SEARCH-PER-MAIL' may be unseen in the dialog example database (e.g., the constraint of 'DA=YN-QUESTION' is negatively correlated to the constraint of 'MG=SEARCH-PER-MAIL'). To generalize the dialog state, the heuristic method first removes the slot flag of 'PER_NAME=1', but Q_{HUC} may still return no examples. However, our proposed method can remove the irrelevant constraint of 'DA=YN-QUESTION' based on their correlation relationships. Consequently, Q_{CBQR} returns some relevant examples.

V. Experiment & Result

A. Experimental Set-up

Our experiments were conducted with a spoken dialog system to allow humans to ask an intelligent robot for information about buildings (e.g., room number, room name, room type) and people (e.g., name, phone number, e-mail address). If the user selects a specific room to visit, the robot takes the user there. For this system, we collected text-based human-human dialogs of about 880 user utterances from 200 dialogs in Korean, which were based on a set of pre-defined



corr(q_1,q_2)	DA='YN-QUESTION'	MG='SEARCH-PER-MAIL'	PER_NAME='1'
DA='YN-QUESTION'	1	-0.0335	-0.0243
MG='SEARCH-PER-MAIL'	-0.0335	1	0.2418
PER_NAME='1'	-0.0243	0.2418	1

q^*	sum of correlation coefficients
DA='YN-QUESTION'	1 + (-0.0335) + (-0.0243) = 0.9422
MG='SEARCH-PER-MAIL'	(-0.0335) + 1 + 0.2418=1.2083
PER_NAME='1'	(-0.0243) + 0.2418 + 1=1.2175

∴ q^*: PER_NAME='1'

Fig. 2. Example of CBQR algorithm

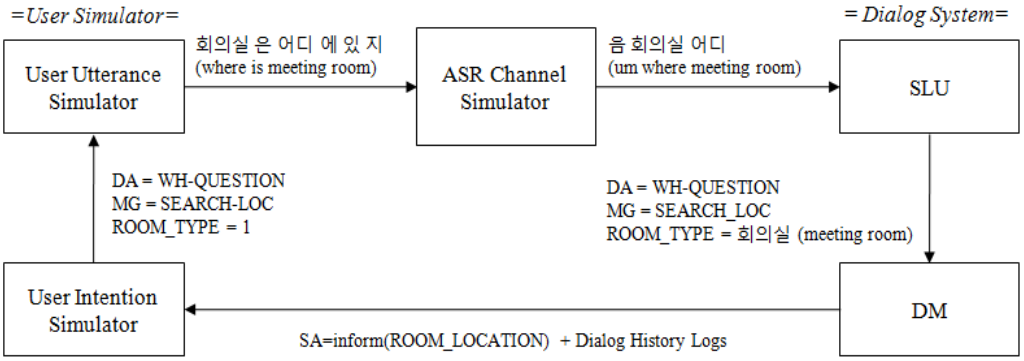


Fig. 3. Simulated environment for automated spoken dialog system evaluation

subjects relating to the building guidance task. Then we implemented the EBDM framework using the CBQR algorithm trained with this corpus.

To automatically evaluate our approach, a simulated environment was developed with a user simulator (both intention simulator and utterance simulator) and an ASR channel simulator (Fig. 3) [11] . This simulator was trained with the same corpus to build the dialog example database. However, the simulator could generate unseen patterns randomly based on probabilistic models. Therefore, our simulator can model speech recognition errors as well as SLU errors at a certain error rate.

To apply the CBQR algorithm, we first obtained a matrix of correlations between all pairs of query constraints. To calculate the correlation matrix, some utterance-level and discourse-level features were manually annotated to each utterance as query constraints for the example search. We used about 100 query constraints (Table II).

Table III illustrates the part-of correlation matrix. It indicates some constraints could co-occur frequently given the dialog state, because some utterances are standardized to represent specific user intentions. For example, in the utterance of "*What is the e-mail address of PER_NAME*", the slot flag 'PER_NAME=1' is correlated with 'MG= SEARCH-PER-MAIL'. However, the slot flag 'LOC_ROOM_NAME=1' is uncorrelated with 'MG=SEARCH-PER-MAIL'. In addition, most utterances of 'MG=GUIDE_LOC' (e.g., "Let's go to the room") is highly correlated to 'DA=REQUEST'.

B. Dialog Simulation Evaluation

Our overall aim is to assess whether our approach facilitates the automation of query relaxation for the example search in the EBDM framework without incurring performance loss. We also hypothesized that using our approach would be robust to speech recognition errors because the co-occurrence of erroneous constraints may seldom exist in the dialog corpus. Therefore, we tested the performance of the dialog models according to different query relaxation algorithms including the heuristic method (as baseline) and our automated method. In the heuristic method, if the exact match returns no examples, the constraint could be first relaxed if it contains the previous information because the system action is usually more dependent on the current input than on the previous input. The only DA constraint was remained at the final relaxation step because it showed the best classification accuracy. The heuristic relaxation was applied at most three times at every turn (Table IV). In addition, we empirically set the correlation threshold of the

TABLE II
LIST OF CONSTRAINT SETS

Types	Constraints	#Size
Utterance-level	Dialog act (DA)	9
	Main goal (MG)	16
	Slot flags	8
Discourse-level	Previous DA	10
	Previous MG	17
	Previous system action	27
	Discourse history vector	16

TABLE IV
SUBSET OF QUERY CONSTRAINTS FOR THE HEURISTIC RELAXATION. ORG DENOTES THE ORIGINAL QUERY FOR EXACT MATCH BY USING ALL QUERY CONSTRAINTS.

Relaxation step	Subset of query constraints
ORG	DA ∧ MG ∧ SF ∧ PrevDA ∧ PrevMG ∧ DHV
R1	DA ∧ MG ∧ SF ∧ PrevDA ∧ PrevMG
R2	DA ∧ MG ∧ SF
R3	DA

TABLE III
PARTIAL EXAMPLES OF THE CORRELATION MATRIX. A HIGHER VALUE MEANS THE GREATER CORRELATION WITH EACH OTHER

q_i	q_j	r_{ij}
DA=REQUEST	MG=GUIDE_LOC	0.6225
DA=YN-QUESTION	MG=SEARCH-PER-MAIL	-0.0335
PER_NAME=1	MG=SEARCH-PER-MAIL	0.2418
LOC_ROOM_NAME=1	MG=SEARCH-PER-MAIL	-0.1149

TABLE V
PERFORMANCES OF USING THE HEURISTIC METHOD (H) AND THE CBQR METHOD (C). THE NUMBERS IN BOLD INDICATE STATISTICALLY SIGNIFICANT IMPROVEMENTS IN PERFORMANCE. † ($p<0.05$) AND ‡ ($p<0.01$) DENOTE THAT THE VALUE IS SIGNIFICANTLY DIFFERENT FROM THE BASELINE BY TWO SAMPLE T-TEST.

WER (%)	Average user turn (per dialog)		Task completion rate		Average query (per turn)	
	H	C	H	C	H	C
0	5.03	**4.96**†	99.22	**99.40**†	1.61	**1.35**‡
10	5.55	**5.46**‡	93.50	**94.18**‡	1.80	**1.46**‡
20	6.20	6.24	87.80	**88.50**†	1.98	**1.55**‡

CBQR algorithm to 0.2, because we did not consider weakly correlated constraints.

To verify our approach, the dialog systems were tested with 20000 simulated dialogs at different word error rates (Table V). The exact match was first done, and if the exact match failed, the partial match was done with relaxed constraints through heuristics (H) and our automated approach (C). As shown in Table V, when speech recognition errors occurred, the use of the CBQR algorithm slightly increased task completion rate, while there was no much difference in average user turn length. In addition, compared to the heuristic method, the average number of queries per turn was significantly reduced. This reduction indicates that the relaxed query generated by the CBQR algorithm could return enough dialog examples to predict the best example with fewer query relaxation steps than used in the heuristic relaxation approach. In conclusion, compared to the heuristic method, our approach improved the performance as well as automatically relaxed failed queries.

VI. CONCLUSION & DISCUSSION

In this paper, we explored how to automatically choose a good query constraint subset using the linear correlations between all pairs of constraints. Although this work is similar to feature selection in machine learning tasks, we think that this work is the first successful attempt at exploiting the constraint-constraint correlations to generalize the dialog state in the spoken dialog systems. In our approach, the most correlated constraint may be automatically selected without needing intervention by human experts. In addition, we think that our approach has two advantages. First, the query relaxation is accomplished using a domain-independent and an automated method using only the correlation matrix. Second, this relaxation strategy can be robust to prior errors because irrelevant constraints can be eliminated based on their correlation coefficients. Therefore, we expect that our approach can be easily applied to find relevant examples for several example-based approaches.

There are several possible subjects for further research on our approach. Firstly, various types should be relaxed for content management, because the proposed method can relax only nominal values. Next, the correlation threshold will be automatically learned to improve our approach by investigating the trade-off between performance and computational efficiency. Finally, we can obtain the correlation matrix using distance metric learning algorithms to estimate more-precise relationships over the input space.

ACKNOWLEDGMENT

This research was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Knowledge Economy (MKE).

REFERENCES

[1] S. Varges, F. Weng, and H. Pon-Barry, "Interactive Question Answering and Constraint Relaxation in Spoken Dialogue Systems," in Proc. of the SIGDIAL Workshop on Discourse and Dialogue, 2006.

[2] I. Muslea, and T. J. Lee, "Online Query Relaxation via Bayesian Causal Structures Discovery," in Proc. of the AAAI, 2005, pp. 831-836.

[3] N. Mirzadeh, F. Ricci, and M. Bansal, "Supporting user query relaxation in a recommender system," *Lecture notes in computer science*, pp. 31-40, 2004.

[4] L. Wen-Syan, K. Candan, Q. Vu *et al.*, "Query Relaxation by Structure and Semantics for Retrieval Oflogical Web Documents," *IEEE Trans. on Knowledge and Data Engineering,* vol. 14, no. 4, pp. 768-791, 2002.

[5] C. Lee, S. Jung, S. Kim *et al.*, "Example-based Dialog Modeling for Practical Multi-domain Dialog System," *Speech Communication,* vol. 51, no. 5, pp. 466-484, 2009.

[6] J. D. Williams, and S. Young, "Scaling Up POMDPs for Dialog Management: The "Summary POMDP" Method," in Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding, 2005, pp. 250-255.

[7] J. Henderson, O. Lemon, and K. Georgila, "Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets," *Computational Linguistics,* vol. 34, no. 4, pp. 487-511, 2008.

[8] F. Lefevre, and R. d. Mori, "Unsupervised State Clustering for Stochastic Dialog Management," in Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding, 2007, pp. 550-555.

[9] M. Hall, "Correlation-based feature selection for machine learning," Ph.D. Thesis, The University of Waikato, 1999..

[10] H. Liu, and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Trans. on Knowledge and Data Engineering,* vol. 17, no. 4, pp. 491-502, 2005.

[11] S. Jung, C. Lee, K. Kim *et al.*, "Data-driven user simulation for automated evaluation of spoken dialog systems," *Computer Speech and Language*, 2009.