

Evaluation of a Similarity-based Approach to Customer-adaptive Electronic Sales Dialogs

Andreas Kohlmaier, Sascha Schmitt, Ralph Bergmann

Artificial Intelligence – Knowledge-Based Systems Group
Department of Computer Science, University of Kaiserslautern
67653 Kaiserslautern, Germany
Phone: ++49-631-205 3362, Fax: ++49-631-205 3357
{kohlma|sschmitt|bergmann}@informatik.uni-kl.de

Abstract. Personalized consultation in large-scale businesses like found in electronic shops is getting more and more important. To provide customers with adequate product information, an automated, dynamic, customer-adaptive dialog has to be carried on that simulates a real sales talk between a sales person and a customer. The task of an electronic sales system is to gain very quickly as much information as possible from the customer. In this context, many approaches for dynamic question selection in automated dialogs only concentrate on dialog length using an information gain measure like the entropy approach. Neither the customers' product knowledge nor the quality of the produced dialogs is taken into account. Here, we present a new measure for question selection in combination with consideration of the costs a question causes dependent on the respective customer class. Detailed evaluations show that this approach produces dialogs that reach the expected retrieval result with fewer questions.

Keywords: customer-adaptive sales dialog, demand acquisition, information gain measure, variance, case-based reasoning, electronic commerce scenario.

1. Introduction

Online customers want personalized advice and product offerings instead of simple possibilities for product search. They need information adequate to their demands instead of pure data [9]. On the Internet, a large number of electronic shops (e-shops) can be found, but most of them only provide the necessary, rudimentary functionality for presenting and selling products. However, following the idea "what the customers do not find, they will not buy", the search for appropriate products in a product database with respect to the customers' requirements has to be well processed. Gaining sufficient information from the customers but also providing them with information at the right place is the key. At this stage, it becomes clear that electronic markets require new ways to appropriately provide consultation in a procedure adapted to their customers. Resulting from this fact, an automated, dynamic,

customer-adaptive communication process is needed that simulates the sales dialog between customers and sales persons.

In this paper, we present a new fundamental mechanism for question selection in such sales dialogs between customers and an electronic sales system. We rather decided for a simple way of carrying on a sales dialog with a customer: the system asks a couple of questions and the customer can choose from a set of appropriate answers to give to the system.

Especially in electronic commerce (EC) scenarios, it is very important to ask as few questions as possible adapted to the customers' knowledge about the product space. It has to be taken into account that online customers are very quickly annoyed and/or bored and the next e-shop is only one mouse click away. That also means that products have to be found as quickly as possible to present them to the customer. Therefore, we integrated the search process into the questioning process.

As a search technology, we used Case-Based Reasoning, CBR. Recently, a couple of CBR approaches have been suggested to adapt automated sales dialogs dynamically to the current sales situation [1,8]. The ideas that can be found have in common that their aim is the reduction of dialog length, i.e. reduction of the number of questions a customer is asked by the sales system. In general, the approaches are based on an information gain measure that is used to select the next attribute to ask which is maximally discriminating the product database, i.e. limits the number of product cases. Amongst several other drawbacks in those approaches, the system-inherent similarity information is neglected and stays unused in this context. Furthermore, question selection should also depend on a probability estimation of the customers' ability to answer this question. To manage these probabilities, we use a Bayesian Network that also learns from the current customer's behavior. To finally select a question to ask, utility values are derived from combination of the probabilities and the (similarity-influenced) information gain.

To evaluate our approach in detail, we established a test scenario that not only examined the length but also the quality of the produced dialogs. We compared the results to an entropy information gain measure used by a commercial retrieval engine. We took a product database with personal computer (PC) systems.

Section 2 describes the principle of a dynamically interpreted dialog strategy and presents several influencing factors of the produced dialog. A question selection strategy based on similarity influence that utilizes the knowledge contained in similarity measures is presented in section 3. Section 4 deals with the necessity to dynamically adapt to the customer during the dialog as not all questions have the same answering cost for everyone. Section 5 presents the comprehensive evaluation of the different influence factors. We end with related work and conclusions as well as an outlook on future work.

2. Dynamically interpreted dialog strategy

In our sense, a dynamically interpreted dialog strategy does not process a previously generated decision tree, but also decides which attribute to ask next during the dialog. This has the dual benefit of being more flexible to adapt to the current customer and

to avoid the construction of an exhaustive decision tree, which can be problematic for unlabelled data (unclassified products) and continuous value ranges. [4,8]

Input: productBase
Output: set of retrieved products

Procedure Dialog (productBase)

candidate_products := productBase;

query := empty_query;

While not **Terminate** do {

 attribute := **Select_Attribute**(candidate_products, query);

 value := **Ask_Question**(attribute);

 query := **Assign_Value**(query, attribute, value);

 candidate_products := **Partition**(candidate_products, query); }

end;

Fig. 1. Algorithm for a dynamically interpreted dialog strategy. (In a meta-language notation.)

Figure 1 presents the principle algorithm for a dialog strategy to be computed at runtime. The products that can be offered by the e-shop are stored in a product database (productBase). Products are described by certain properties which we call attributes. The strategy starts with an empty problem description (query), i.e. what the system has found out about the customer's product wish so far, and chooses a first question (for such an attribute's value) according to the question selection strategy. Depending on the answer to the posed question, the set of candidate products is reduced and the process is iterated until a termination criterion is fulfilled. Three different aspects can be identified that influence the dialog strategy:

1. The **question selection strategy** determines which attribute to ask next and influences the *dialog length*. A good questioning strategy leads to minimal dialogs with optimal results.
2. The **termination criterion** determines when enough information has been gathered. It therefore influences the *quality of the result* and the dialog length. A perfect termination criterion should continue asking questions until the optimal result is reached. Since it is not known in advance what the optimal solution is, several possible termination criteria can be examined. E.g., Doyle & Cunningham [1] suggest continuing asking questions until a manageable set of n cases remains (e.g., $n = 20$) or all attributes are asked. A more suitable way for EC is to check if the expected information gain for all remaining attributes falls below a given threshold.
3. The **partitioning strategy** is used to reduce the search space to the best matching candidate products. However, in this paper, the influence of partitioning will not be further investigated.

Section 5 gives experimental data that measures the influences of the described factors on the length and quality of the produced dialogs.

3. A similarity influence measure for question selection

Questions asked to the customer should be selected on the basis of how much information they contribute to select possible products from the product database. Most question selection strategies evaluate the information gain for a given attribute on the basis of the distribution of attribute values to distinct classes. The most commonly known strategy is the measure of expected information gain introduced by Quinlan [4] in the ID3 algorithm for the construction of decision trees.

A different approach that is better tailored to deal with online product sales is to select attributes not on the basis of their information gain but on the basis of the influence of a known value on the similarity of a given set of products. In an online shop it is desirable to present the customer a selection of products most similar to her/his query. It is therefore a reasonable strategy to first ask the attributes that have the highest influence on the similarity of the cases (products) stored in the product database.

A way to measure the influence on similarities is to calculate the variance Var of similarities a query q induces on the set of candidate products C :

$$Var(q, C) = \frac{1}{|C|} \cdot \sum_{c \in C} (sim(q, c) - \mu)^2 \quad \text{(Variance of Similarities for a Query)}$$

Here, $sim(q, c)$ denotes the similarity of the query q and the case c , μ denotes the average value of all similarities.

When asking a question, the assigned value is not known in advance. It is therefore necessary to select the attribute only on the expected similarity influence $simVar$, which depends on the probability p_v that the value v is chosen for the attribute A :

$$simVar(q, A, C) = \sum_v p_v \cdot Var(q_{A \leftarrow v}, C) \quad \text{(Expected Similarity Influence of an Attribute)}$$

$Var(q_{A \leftarrow v}, C)$ defines the similarity influence of assigning a value v to an attribute A of the query q . To simplify the computation of $simVar(q, A, C)$ it is possible to consider only the attribute values v that occur in the product set C . Then, the probability p_v for the value v can be estimated by the sample of products in C , i.e. $p_v = |C^v| / |C|$.

In a dialog situation, the attribute with the highest expected similarity influence on the set of candidate products is selected. This strategy leads to the highest increase of knowledge about similarity thereby faster discriminating the case base in similar and dissimilar cases.

An aspect that is not considered by other proposed approaches for question selection strategies is the different cost of questions estimated by the probability that the customer can answer the question. The ensuing section deals with this issue.

4. Answering cost estimation

Current approaches for EC implicitly assume that every question has the same cost. This assumption is fairly accurate as long as every customer can and is willing to answer every posed question. But in a real world EC scenario, it is quite possible that a customer does not answer a question, either because s/he does not care about the proposed attribute or s/he does not understand the meaning of the attribute. Traditional attribute selection strategies can be misleading in this situation because they do not take into account that asking a question may not result in the assignment of a value to an attribute. It is therefore necessary to model in more detail the possible outcomes of asking a question and to define a *utility* for the different outcomes. During the dialog situation, the question with the highest expected utility will be asked. An additionally very important factor is the customers' degree of satisfaction during the dialog. According to Raskutti & Zuckerman's nuisance factor [3], we introduced a satisfaction level to mirror this aspect. This level is decreased dependent on the questions posed and the customers' action respectively. Usually, a customer will not answer any desired number of questions, especially if the questions are not understood or s/he does not care about the attribute asked.

The *expected utility* EU of an action A^1 with the possible results $Result(A)$ is defined by the probability $p(a|E)$ that a is the outcome of A based on the current knowledge E of the world and the utility of the outcome a , as suggested by Russell & Norvig [6]. Utilities model a preference relation between the different outcomes of A . An outcome a with a higher utility is preferred to an outcome with lower utility. The expected utility of an action A can be defined as:

$$EU(A | E) := \sum_{a \in Result(A)} U(a_A) \cdot p(a_A | E) \quad \text{(Expected Utility of an Action)}$$

Table 1 summarizes the possible outcomes of questions and their utility.

To assign exact numerical values to the utility of each outcome depending on the expected information gain (*info*) of the attribute, the following function

$U: Result(A) \rightarrow [-1,1]$ is used:

$$U(a_A) := \begin{cases} info(A) & \text{if } a_A = \text{"answered without help"} \\ info(A) - d & \text{if } a_A = \text{"answered with help"} \\ 0 & \text{if } a_A = \text{"don't care"} \\ 0 - 2d & \text{if } a_A = \text{"don't understand"} \end{cases}$$

Here d denotes the penalty for lost information because a question cannot be asked because of the decrease in customer satisfaction. To assign a value to d , it is necessary to assess the *a priori* information gain of the chance to ask a question. Since the information gain of future questions is not known in advance, a plausible estimation is

¹ In our case, there is only one action A for an Attribute A , namely asking the attribute's value. E.g., Shimazu [7] suggests "navigation by proposing" as further action.

to give d half the information gain of the current question. Of course, it is conceivable to choose d differently, e.g., dependent on each one of the customer classes. More utility functions have to be investigated in the future.

Table 1. Possible outcomes of posing a question and their expected utility.

Event	Description	Effect	Utility
<i>answered without problems</i>	the question was answered without aid of the help function	the attribute is assigned a value, information is gained	high, depending on the gained information
<i>answered with help</i>	the question was confusing, but the customer managed to answer the question after studying the built-in help system	the attribute is assigned a value, information is gained, the customer is bothered	medium, depending on the gained information
<i>don't care</i>	the customer understands the meaning of the question, but its result is of no importance to her/him	no value is assigned, information on the importance of attributes is gained	none
<i>don't understand</i>	the customer could not answer the question	no information gain, the customer is frustrated	negative, because of information loss

To compute the overall utility of a question, the probabilities $p(a|E)$ are needed. These probabilities differ for every customer and strongly depend on her/his background knowledge of the given domain. We use a Bayesian Network (see, e.g., [3]) to assess the customers' background knowledge.

5. Evaluation of dynamic dialog strategies

To test the different aspects of our approach for a dynamic dialog system, we used a domain of 217 cases describing PC systems. Each case consisted of 28 attributes ranging from more generally known attributes to highly technical ones. The cases were generated by a PC configuration system [8].

5.1 Test environment

We employed the *leave one out* strategy for each test 200 times. A single case was removed from the case base and used as the reference query, describing the completely known customer's demands. This reference query was used for retrieval on the case base and returned the ideal result, i.e. the best possible result when all information is available. The result consists of an ordered list of the 10 cases with highest similarity to the reference query. We used a *customer agent* to simulate the behavior of a real customer. The customer agent was repetitiously asked questions by the system and supplied one by one the attribute values of the reference query. We implemented two different kinds of agents, an ideal customer agent that can answer every question and a real life customer agent that can only answer questions with a certain probability. After each question, a retrieval with the partially filled query was performed and this result was compared to the ideal retrieval result of the reference query. The result of the retrieval was considered successful if the three best cases of

the current retrieval result could be found amongst the five best cases of the ideal retrieval result (reference retrieval). In our experiments, we measured how many dialogs out of the total number of 200 were successful for a given dialog length. This result is a good measure for the quality of the dialog strategy as it measures how quality of the result increases with the length of the dialog.

Here, it should be remarked that data gained from real customer behavior would be more meaningful. However, as a first stage, it was our primary goal to investigate in the possibilities of our new approach. So, the simulated customer represents an upper bound of what can be reached by customer-adaptiveness.

5.2 Evaluation of the attribute selection strategy

The most significant influence on the quality of the dialog lies in the attribute selection strategy. In the first test, we compared the *similarity variance measure* *simVar* to an *entropy-based* one as a traditional representative of an information gain measure. We used an ideal customer agent that can answer every question. The second test shows how these strategies behave in a real world environment using the real world customer agent and shows the benefit of considering question-answering costs in real world situations.

To avoid the influence of the termination criterion, we let the questioning continue until all questions had been asked. It was recorded how many out of the total of 200 performed dialogs were successful at a given dialog length.

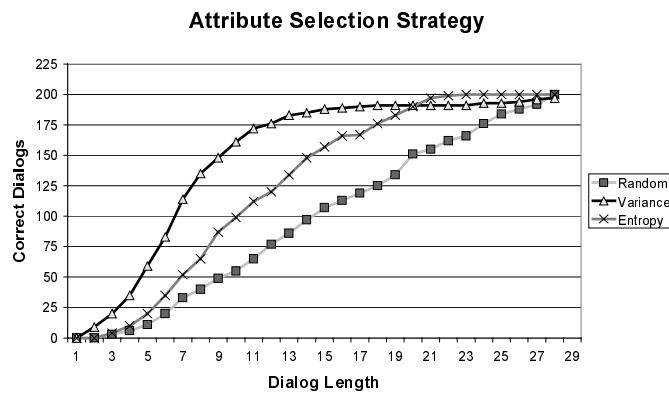


Fig. 2. Comparison of *random*, *simVar*, and *entropy* attribute selection strategies.

5.2.1 Test 1: Ideal customer agent

The chart in Fig. 2 depicts the result for an ideal customer agent acting in a sales dialog environment for *random*, *similarity variance* (*simVar*), and *entropy* attribute selection strategies. Since the ideal customer is able to answer every question, adaptation is not necessary.

Maximizing the variance is generally a good strategy to separate the best cases, however, it is sometimes necessary to tolerate decrease in variance when most cases

have already been excluded from the candidate set. While a strategy like ours that strictly maximizes the variance avoids such questions and leads to a leveling of the curve, such a heuristic is nevertheless justified by the maximum likelihood principle.

Compared to the random method which shows only a linear progression and the steeply rising entropy method, *simVar* is the best strategy.

5.2.2 Test 2: Real world agent

The second series of tests simulated a real world scenario of an e-shop. Therefore, it was executed with customer agents that could not answer all questions. Each agent simulated a customer with certain knowledge of the problem domain. That means an agent either had expert, intermediate, or little (beginner) knowledge of PCs. When an agent was asked a question it could answer the question with a certain probability depending on its knowledge level. This probability was looked up in the conditional probability tables stored with each question. This implicitly assumes that our Bayesian Network was optimally trained as the simulated customer behaved exactly as modeled in the network. So, the results also represent an upper bound of what can possibly be gained considering question answering costs.

We tested several different strategies to adapt to the customer to see how well they perform for each customer class.

- **Pure information gain measure without customer adaptation.** This strategy ignores the customer knowledge and selects the question with the highest information gain. No customer adaptation is done.
- **Highest probability of an answer.** This strategy ignores the information gain of an attribute and selects those questions that have the highest chance to be answered.
- **Highest utility.** The utility measure gives a detailed assessment of what can be gained (or lost) by asking this question.

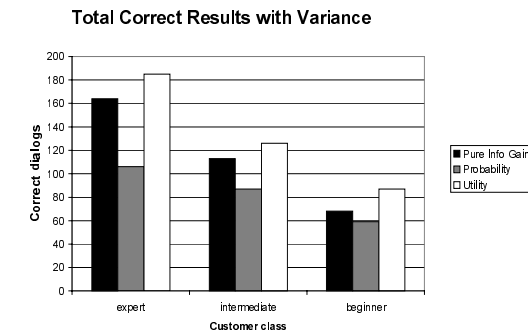
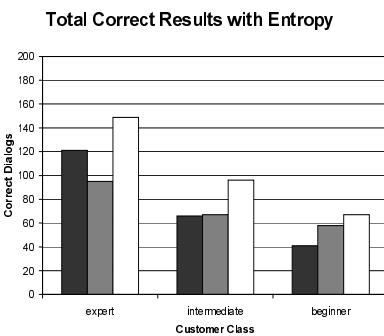


Fig. 3. Series No. 1: Entropy strategy tested with a real world agent.

Fig. 4. Series No. 2: *simVar* strategy tested with a real world agent.

A separate test series was carried out for each customer class and each information gain measure to see how well the different customer adaptation strategies perform.

Fig 3 shows that the entropy strategy can be drastically improved by considering utility. This is the case, because questions preferred by the entropy strategy are those with the greatest distribution of values, which can be difficult to answer. This is also reflected by the fact that the probability approach produces better results than the pure entropy one for beginner customers.

The increase for the variance strategy, as depicted in Fig. 4, is not so high. This can be explained by the fact that the variance strategy already prefers questions that are easy to answer. The questions chosen by *simVar* have the highest weight in the similarity calculation. These are most likely the key features of the product (such as price or processor speed), which can easily be understood by most customers. This also justifies the poor performance of the pure probability strategy.

Analyzing Figures 4 and 5, it can be seen that all *simVar* strategy methods clearly reach better results in terms of correct dialogs than their entropy competitors.

6. Related work

A couple of commercial systems providing dialog functionality for EC systems are on the market but none of them follows an approach like ours. Examples for such systems are orange and its former version CBR-Works from tec:inno - empolis², both with an entropy-based approach. Furthermore, there is the Pathways system from IMS MAXIMS³ that only allows defining static decision trees. The often cited PersonaLogic⁴ system asks a large number of questions, not adapting to the customer at all. However, the search process is integrated in the dialog filtering the current candidate cases.

From the scientific point of view, several publications have been dealing with the topic of attribute selection in the field of CBR. Their origin is in diagnosis systems. The approaches have in common that they are based on entropy information gain measures and decision tree learning. None of them neither considers the described costs nor the quality of the produced dialogs. Currently most relevant to our approach is the work of Doyle & Cunningham [1], but their approach bases on a classification of the cases. Shimazu [7] proposes navigation methods in combination with an attribute selection strategy. Göker & Thompson [2] follow an unconventional way by a rule-based approach with dialog operators.

7. Conclusions and future work

We developed a new approach for attribute selection especially tailored to EC scenarios and compared it to the traditional entropy-based proceeding. It turned out that sensible dialogs are generated by always selecting the combination of the most important and easiest to answer question.

² <http://www.tecinno.com/>

³ <http://www.cykopaths.com/>

⁴ <http://www.personalogic.com/>

We have not yet investigated in the possible dependency between our approach and the complexity of the domain used for our tests. Factors of interest are the distribution of cases in the case base, the knowledge contained in the similarity measures, and the existence of attributes that require different expertise to be answered.

A couple of open questions have already been raised in this paper. They are currently under investigation or have to be investigated respectively in the near future. Here, we only want to hint on a few more aspects. Our approach does not guarantee an optimal or logical ordering of questions from a customer's point of view. This issue could be addressed by another influence factor for our utility calculation, which we get, e.g., out of an extension of our Bayesian Network. An important next step will be the training of the Bayesian Network with real customer data from an e-shop. Therefore, we have to optimize the processing of *simVar* to deploy it in live scenarios.

References

- [1] M. Doyle, P. Cunningham. (2000). A Dynamic Approach to Reducing Dialog in On-Line Decision Guides. In: E. Blanzieri, L. Protinale (Eds.): *Advances in Case-Based Reasoning. Proceedings of the 5th European Workshop on Case-Based Reasoning, EWCBR 2000*, Trento, Italy. LNAI 1898, Springer.
- [2] M. H. Göker, C. A. Thompson. (2000). Personalized Conversational Case-Based Recommendation. In: E. Blanzieri, L. Protinale (Eds.): *Advances in Case-Based Reasoning. Proceedings of the 5th European Workshop on Case-Based Reasoning, EWCBR 2000*, Trento, Italy. LNAI 1898, Springer.
- [3] J. Pearl. (1988). *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann Publishers.
- [4] J. R. Quinlan. (1993). *C4.5 Programs for Machine Learning*. Morgan Kaufmann Publishers.
- [5] B. Raskutti, I. Zuckerman. (1997). Generating Queries and Replies during Information seeking Interactions. In: *International Journal of Human Computer Studies*, 47(6).
- [6] S. Russell, P. Norvig. (1995). *Artificial Intelligence: A Modern Approach*. Prentice Hall International Editions.
- [7] H. Shimazu. (2001). ExpertClerk: Navigating Shoppers' Buying Process with the Combination of Asking and Proposing. To appear in: *Proceedings of the 17th International Joint Conference on Artificial Intelligence, IJCAI-01*, Seattle, Washington, USA.
- [8] A. Stahl, R. Bergmann. (2000). Applying Recursive CBR for the Customization of Structured Products in an Electronic Shop. In: E. Blanzieri, L. Portinale (Eds.): *Advances in Case-Based Reasoning. Proceedings of the 5th European Workshop on Case-Based Reasoning, EWCBR 2000*, Trento, Italy. LNAI 1898, Springer.
- [9] M. Stolpmann, S. Wess. (1999). *Optimierung der Kundenbeziehung mit CBR-Systemen*. Addison-Wesley.