SemDial 2011: Proceedings of the 15th Workshop on the Semantics and Pragmatics of Dialogue, pages 37–46.
Los Angeles, California, 21–23 September 2011.

A decision-theoretic approach to finding optimal responses to over-constrained queries in a conceptual search space

Anton Benz and **Núria Bertomeu** and **Alexandra Strekalova**
KomParse Project*
Centre for General Linguistics (ZAS)
Schützenstrasse 18, 10117 Berlin

Abstract

The problem of how a recommender system should react to over-constrained queries has often been discussed. A query is over-constrained if the stated preference combination cannot be satisfied by any item in the database. We address the generation of cooperative responses to over-constrained queries in the context of a conversational recommender system embodied by an artificial agent whose task it is to help customers furnishing their living-room. The solution proposed is designed for preference statements of qualitative nature, and is aimed at avoiding arbitrary weightings of preferences. It combines Decision Theory with a cognitive model of multi-dimensional concept representation, *conceptual spaces* [8], which allows the assessment of preference according to distances in the cognitive space.

1 Introduction

This paper grows out of the KomParse project which aims at the design of non-player-characters (NPC) with natural language dialogue capabilities for virtual environments. More specifically, we have developed an artificial sales agent / interior designer who helps customers furnishing their virtual living room. In this scenario, the NPC embodies a conversational recommender system. Here follows an example dialogue, taken from a corpus of NPC-human dialogues in the furniture sales scenario obtained by a Wizard-of-Oz experiment [3]:

(1) USR.1: And do we have a little side table for the TV?
NPC.1: I could offer you another small table or a sideboard.
USR.2: Then I'll take a sideboard that's similar to my shelf.
NPC.2: What about this one?
USR.3: No, that doesn't fit in here.
NPC.3: Here is another one. Do you like it?
USR.4: Is there a black or white sideboard?
NPC.4: No I'm afraid not, they are all of light or dark wood.
USR.5: Ok, then I'll take this one.
NPC.5: All right.

Two requirements are of special interest to us: the ability of the NPC to understand and represent the user's preferences, and the ability to generate answers that optimally contribute to the satisfaction of these preferences. An important problem is the generation of such optimal responses in a situation in which the constraints imposed by the user's preferences cannot be satisfied by any object in the database. This situation is commonly called retrieval failure and the query leading to it is said to be over-constrained. Turns NPC.1 and NPC.4 of (1) are examples of optimal responses to retrieval failures.

The NPC can react to an over-constrained query in different ways. The following example shows some of them:

(2) USR.1: Let me see a modern one ... If it's possible a yellow one, please.

*The research reported in this paper has been conducted as part of the project KomPARSE, carried out in cooperation by the Zentrum für Allgemeine Sprachwissenschaft (ZAS) and the Deutsches Zentrum für Künstliche Intelligenz (DFKI) from June 2008 until June 2011. The project is funded by the ProFIT program of the Investitionsbank Berlin and the European Regional Development Fund.

NPC.1a: I don't have something like that.
NPC.1b: I don't have something like that, but may be you will like this one. (*shows object*)
NPC.1c: (I don't have something like that, but) I can offer you a modern sideboard in white or a vintage one in yellow.

In (a) the NPC just informs the user about the unavailability of the desired type of furniture. In (b) he additionally shows an alternative object. In (c) the NPC additionally proposes two alternatives that are similar to the requested object. For each alternative, one of the requested characteristics is kept, while the other is relaxed and a new value which is similar to the requested one is proposed. Clearly, (a) is the least informative response, and for (b) the NPC has to have good reasons to believe that the selected object is best fitting to the user's preferences. In (c) the NPC is uncertain about which of the alternatives found better fulfills the preferences of the user, so he decides to present at least some of them. In this paper we will present a general approach to finding optimal alternatives to an over-constrained query. Additionally, we will address the generation of answers of type (c).[1]

The dialogue course of action is the following: first, the system asks the user an open question about his preferences and the user provides a property-value combination as an answer. Alternatively, the user may request a property-value combination without having been asked before. Next, the system performs a database search for objects exhibiting the property-value combination demanded by the user. If the search fails to return some item, the system looks for alternatives to propose. As in (c), the system may propose a set of alternative property-value combinations, from which the user may choose one. An object exhibiting the chosen property-value combination is then shown to the user.

In human-to-human communication, responses do not only communicate their literal content but also additional implicatures. Assuming that the sales agent is maximally cooperative, an answer like (a) implies that there is no good alternative unknown to the user. Showing an object as in (b) does not carry linguistic implicatures, but one might infer that the sales agent believes that the selected object is best fitting to the user's preferences, although there may be other interesting alternatives. Finally, (c) carries the implicature that the modern sideboard in white and the vintage sideboard in yellow are, to the speaker's best knowledge, among the best alternatives which he can offer. There is also a strong tendency to understand this alternative exhaustively, i.e. as meaning that all other alternatives, if they exist, are even more remote from the user's preferences. In order to maintain a human-like appearance, the responses generated by the NPC must vindicate these implicatures.

Our theoretical model for determining optimal responses to over-constrained queries handles not only simple preference statements such as '*I want a yellow sideboard,*' but also more demanding ones, such as the *similarity* requirement in (1) '*a sideboard that's similar to my shelf.*' We will address this problem in Section 5. Our representation will be based on a *multi-attribute utility analysis* [10]. The main theoretical problem is the search for optimal alternatives, if preferences cannot be matched. This becomes a problem as the preference statements of the user, in general, underdetermine their preferences over available database entries. We approach this problem by stating preferences in the furniture sales scenario as preferences over property combinations in a *conceptual space* [8]. We show how the natural similarity relations on conceptual spaces can be of crucial use in the search for alternatives.

The proposed model is of interest for a system in which no user model containing information about the values preferred by the user for the different attributes and about the relative importance of those is available. In such situation the design has to rely on a priori knowledge about domain properties. We show how for this problem a combination of conceptual spaces with a multi-attribute utility analysis can be used for finding optimal responses. In Section 2, we introduce our theoretical model for the retrieval of optimal alternatives, and show how to apply it with an extended example in Section 3. In Section 4, we discuss related work. In section 5, we propose several techniques to reduce the retrieval set if it becomes too large. Finally, in Section 6 we summarize and conclude.

[1]The choice of the type of response is relegated to a separate content planning module which we will not describe in this paper. The content planning module does not only decide which alternatives to present, but also how to present them, e.g. whether to show an object, as in (b), or request information about the user's preference over a set of property-value combinations, as in (c).

2 The framework

In the dominant BDI (belief, desire, intention) framework of modal logic, a statement as '*I would like to have a* purple leather sofa' would receive a representation similar to $\Box\,\exists x(have(I,x) \wedge sofa(x) \wedge purple(x) \wedge leather(x))$, where $\Box$ is a modal operator for *desire* such that $\Box\varphi$ is true iff φ is true in all desired worlds [11]. Modal logic representations are plagued by a number of well-known paradoxes. One especially relevant to our scenario is Ross' s paradox [16], a variant of which is the inference from '*I want that the letter is mailed*' to '*I want that the letter is mailed or burned*' which is valid in a standard BDI framework. Hence, also the inference from '*I want a purple leather sofa*' to '*I want a purple or green leather sofa*' is valid. We therefore opted for a framework based on *multi-attribute utility theory* (MAUT) [10].

In Decision Theory, preferences are represented by utility functions which map the possible outcomes of decisions, in our case the objects of the catalogue, to real values. If these preferences are only stated qualitatively, then only the fact that some outcome is preferred over another is known but not the degree of the preference. Arguably, in Example (1), all preference statements are qualitative in nature. Ceteris Paribus (CP) nets [5, 4] allow the representation of qualitative preference statements as a directed graph. CP-nets have recently been proposed as a framework for the semantics of natural language statements about preferences [2]. The representation of preferences in CP-nets is based on MAUT. If we can assume that preferences over outcomes of decisions only depend on a finite number of attributes $\{F_1,\ldots,F_n\}$, then the pre-order $\preceq$ over outcomes can be represented by a pre-order over n-tuples $\{a_1,\ldots,a_n\}$ of values a_i for attributes F_i. In our scenario, the attributes are properties like colour, material, and size. CP-nets allow the representation of conditional preference statements of the form: if a_1, then a_2 is preferred over a'_2. This statement receives a *ceteris paribus* interpretation: a_2 is preferred over a'_2 given a_1 if the value of all other a_i are equal. CP-nets do not allow the representation of probabilities or gradual preference statements.

As mentioned before, the preference statements in (1) are qualitative and not graded. Nevertheless, CP-nets turn out to be unsuitable for a number of reasons. In general, a user interacting with an NPC will not provide a complete characterization of his preferences. For example, if the customer says that he wants to have a purple sofa, then we can infer that red or yellow is less desired, but we cannot logically infer that a red sofa is more desired than a yellow sofa. However, we need this information for proposing alternatives. As a detailed preference elicitation is not viable, the CP-net resulting from utterance interpretations will leave the preferences highly underspecified, and no useful inferences about the user's relative preferences for alternative values can be drawn. In order to facilitate such inferences, we made use of the natural similarity measures of the property domains as they are represented in conceptual spaces [8]. We therefore based our representation of preferences more directly on MAUT by making use of real-valued preference functions on conceptual spaces. In the next section, we explain our representation of preferences and the retrieval of optimal alternatives according to those.

We make the simplifying assumption that the customer's preferences can be represented by an *additive multi-attribute utility function* [10]. This means that each database object a can be identified with a sequence of attribute values $\langle a_1,\ldots,a_n\rangle$ such that the customer's utility function F can be decomposed into the sum of his preferences over the different attributes which in turn can be represented by a non-negative real valued function F_i for the i'th attribute:

$$F(a) = F_1(a_1) + F_2(a_2) + \ldots + F_n(a_n) \qquad (2.1)$$

A consequence of this representation is that the preferences over the i-th dimension satisfy a ceteris paribus condition. This means, if $F_i(a_i) > F_i(a'_i)$, then a_i is ceteris paribus preferred over a'_i. The customer's preference statements reveal a desired combination of attribute values $\langle a'_1, a'_2, \ldots, a'_m\rangle$ where not all possible attributes need to receive a value. Hence, the sales agent cannot be sure that the stated attribute-value combinations are an exhaustive list of all relevant attributes, but, for the specific answering situation which we are interested in, he can assume that only those attributes count which are explicitly mentioned. The utility function F can be further constrained by dividing the attributes into hard and soft attributes. For example, when the customer states that he wants a *purple leather sofa*, we can assume that [TYPE = sofa] is a hard constraint, and that the values for COLOUR and MATERIAL define soft constraints. Hence, searching for optimal alternatives is equivalent to a constraint optimization problem for which a database object a has to be found which satisfies all hard constraints and optimizes $F(a)$, where F is a sum of the utilities $F_i(a_i)$ for soft attributes i. The main problem to be solved is how to optimize $F(a)$ without actually knowing F.

At this point, we can exploit the geometrical structure of the colour space. We know that red is closer to purple than yellow is. If we assume that the preferences decrease with increasing distance, we can infer that red is preferred over yellow. This consideration can be generalized to other attributes if the respective domains come with a natural distance measure. The customer's preference statements then define a *target t* in a conceptual space, and we can assume that, if d_i measures the distance between two values of attribute i, then

$$d_i(t,a_i) < d_i(t,a'_i) \Rightarrow F_i(a_i) > F_i(a'_i). \tag{2.2}$$

This still only provides a weak characterization of the utility function F, as we do not know e.g. the value of differences $F_i(a_i) - F_i(a'_i)$ or the relative weight of the different attributes, i.e. $F_i(a_i) - F_j(a'_j)$. Nevertheless, we have enough information to solve the constraint optimization problem. The solution is to provide an answer which is independent of the remaining utility functions.

From now on, we assume that the desired target is an element in a conceptual space, and that for each dimension i of this space the distance of the values a_i from t can be measured by a measure function d_i. Condition (2.2) entails that objects which are closer to the target are preferred. As we are minimising the distance, it is easier to think of F as a *penalty* function, i.e. a utility function for which lower values correspond to more preferred values. This entails that F has to be minimized, and, in particular, that $d_i(t,a_i) < d_i(t,a'_i) \Rightarrow F_i(a_i) < F_i(a'_i)$. As the set of database entries is finite, we can order all values of the i'th dimension according to increasing distance. We can identify them with a set E_i of natural numbers, and the search space with the product $\prod_i E_i$ which may contain more elements than the database. For this search space, we can reduce the problem of finding an optimal proposal of alternatives to a purely geometric problem:

Theorem 1 *Let $(E_i)_{i=1}^n$ be a sequence of sets of natural numbers, $E = \prod_i E_i$, and $e \preceq e' :\Leftrightarrow \forall i\, e_i \leq e'_i$. Let $D \subseteq E$ and $e = (e_i)_{i=1}^n \in D$. Then the following conditions are equivalent:*

1. *e is a $\preceq$-minimal element of D.*
2. *e is an element of the set*

$$K = \{e \in D \,|\, \forall e' \in D : \exists i\, e'_i < e_i \rightarrow \exists j : e_j < e'_j\}.$$

3. *There are functions $F_i : E_i \rightarrow \mathbb{R}_0^+$, $i = 1\ldots,n$, such that*
   (a) *$\forall n,m \in E_i : n < m \rightarrow F_i(n) < F_i(m)$,*
   (b) *and*

$$\sum_{i=1}^n F_i(e_i) = \min_{e' \in D} \sum_{i=1}^n F_i(e'_i)$$

The proof is straightforward. D represents the set of database objects. K is the set of all objects which are such that if there is an object e' which is closer to the target in one dimension, then there is at least one other dimension in which e' is farther from the target. Being an element of K is equivalent to being a $\preceq$-minimal database element.[2] The elements of K are called *Pareto efficient*, or the *efficient frontier* in multi-attribute utility theory [10, p. 70]. The theorem says that whatever the actual preferences of the customer are, as long as they satisfy condition (3a), the set K will contain at least one object which optimally satisfies them. And conversely, if e is an element of K, then there exist preferences of a possible customer for which e is optimal.

The theorem is applied as follows: we divide each dimension in the conceptual space into a finite number of intervals. All the items located in the same interval are treated as equally distant from the target. Thereby, the conceptual search space becomes isomorphic to a product space $E = \prod_i^n E_i$.[3] The elements of E represent n-dimensional *cubes* in the conceptual space. For E, it is a purely geometrical problem to determine K. Each element in K is Pareto efficient. The user chooses one of these cubes. It can be expected that this cube contains a database element which optimally satisfies his preferences.

[2] The condition that the utility function F is *additive* is not really necessary. It is only needed that $e \preceq e' \Leftrightarrow F(e) \leq F(e')$.

[3] More precisely, we first assign to each element in the conceptual space a vector which represents its distance from the target t which was defined by the user's preferences. Hence, we have to assume that the conceptual space is endowed with a suitable vector space metric. This is stronger than the conditions formulated by [8], but it is in line with formalization in the AI literature, see e.g. [1, 15].

The dialogue between the customer and the sales agent can be seen as a joint search for an optimal object in the database. We can conceptualize the situation after a failed search of the database as a game in which the NPC first provides more information about the available objects, and the user then communicates his preference among these available alternatives[4]. The exchange is successful if the new preference combination is the best one which can be satisfied[5]. By presenting K to the customer, it is guaranteed that, whatever the preferences of the customer are, at least one element of K is optimal for him. It can be shown that the presentation of K is the optimal choice for the NPC if the costs of verbally presenting K are negligible. If the goal is to find the best liked object in the catalogue, not just choosing some object, that the user is aware of the available alternatives gives us a guarantee of task completion. Moreover, the dialogue is more efficient, since the user requests unavailable property combinations less often. However, sometimes it is not the case that the costs of verbally presenting K are negligible, and a subset of elements of K has to be selected for presentation. In section 5, we will discuss several approaches to non-arbitrarily choosing a subset of K for verbal presentation.

3 Example

To illustrate the retrieval of optimal alternatives we consider the following example:

(3) USR: I would like to have a *purple leather* sofa.
NPC: I'm afraid we don't have a purple leather sofa, but I can show you a *purple fabric* sofa or a *black leather* sofa.

We assume that the database search returns no result for the stated preferences. How can the NPC generate his answer? Following the framework presented in section 2, the stated properties first are used to define a *target* in a conceptual space which we represent as a feature structure:

$$\begin{bmatrix} \text{COLOUR} & \text{purple} \\ \text{MATERIAL} & \text{leather} \end{bmatrix}$$

As the target only defines values for material and colour, we can assume that the relevant conceptual space is defined by these properties. Gärdenfors [8] distinguishes between *properties* and *concepts*. Properties are defined by a combination of attributes which cannot be attributed to an object independently of each other, i.e. if one attribute has a value, then all other attributes defining the property must have a value. *Colour* is a property which can be described, e.g. in the Hue Saturation Value (HSV) colour model, by hue, saturation and value, three attributes which define the dimensions of a vector space. The HSV value of each colour term is specified in the knowledge base. This information is used for defining the specific HSV value of the target object.

In the next step, the colour space has to be divided into a finite set of colour values which are treated as equally distant from the target value. For simplicity, we assume here that the corresponding equivalence classes partition the colour domain into a set of intervals. The threshold values for the intervals are determined by comparing the distance between shades of the target colour (e.g. purple and amethyst), colours which are neighbours of the target colour on the colour wheel (e.g. purple and blue), and complementary colours, which lie opposite the target value on the colour wheel (e.g. purple $\rightarrow$ yellow). The remaining colours were collected in an interval between the neighbours and the complementary colours. The result is shown in Table 1[6].

Distance	Equivalence class
$t < 100$	0
$100 \leq t < 200$	I
$200 \leq t < 350$	II
$350 \leq t < 550$	III
$550 \leq t$	IV

Table 1: Intervals defining the equivalence classes of the colour dimension.

The second property specified by the customer is the material. The knowledge-base contains information about five attributes such as organic/non-organic,

[4]In the case that there is only one available alternative this will be presented directly.

[5]However, it might happen that once the items that exhibit the preference combination have been seen, the user resorts to a different preference combination because he likes none of the objects. This brings nevertheless the task forward, since the user further adapts his preferences to the available choices.

[6]The values I-IV are the numbers of one dimension in the product space $\prod_{i=1}^{2} E_i$ from Theorem 1.

softness, robustness, see Table 2. These attributes have binary values and define together a five-dimensional space.

Material	organic	rough	soft	robust	cold
Leather	1	0	1	0	0
Fabric	1	1	1	0	0
Plastic	0	0	0	1	0

Table 2: Material properties specified in the knowledge-base.

In analogy to the colour property, the material property is divided into intervals. The distance between the target value and the other materials can be defined by the number of dimensions for which the materials share the same value. For example, fabric is more similar to leather than plastic because leather and fabric share more values: leather and fabric share four values, whereas leather and plastic share only two values. In general, material can differ with respect to all five properties. If all five values are identical with the target value, the material can be assigned to equivalence class 0. If all the values are different from the target value, then it is assigned to equivalence class V. The same principle holds for all intermediate classes.

Let us assume that there are five sofas in the database, which are specified for material and colour, both in HSV format and with the corresponding natural language term (see Table 3).

Object	Properties	
Sofa_Alatea	COLOUR	red
	MATERIAL	fabric
Sofa_Consuelo	COLOUR	yellow
	MATERIAL	fabric
Sofa_Grace	COLOUR	airForceBlue
	MATERIAL	fabric
Sofa_Nadia	COLOUR	black
	MATERIAL	leather
Sofa_Isadora	COLOUR	amethyst
	MATERIAL	fabric
having larger		

Table 3: Catalogue items specified for colour and material

In the next step, the colour and material values of each sofa are assigned to an equivalence class in the corresponding dimension. For example, the distance between the desired colour purple and the colour of sofa Alatea is 319. This value puts the later into equivalence class II. The distance between purple and yellow is 420, which puts sofa Consuelo into equivalence class III. The distance between purple and amethyst is 192, which puts sofa Isadora into equivalence class I. Therefore, if we only considered the colour of sofas Alatea, Consuelo and Isadora, sofa Isadora would be the candidate which fits best the customer's preferences.

In our example only sofa Nadia is made of leather, the value desired by the customer. Therefore, it is assigned equivalence class 0. All other sofas have the value fabric. Fabric shares with leather all values except roughness/smoothness, so it is assigned equivalence class I. Table 4 shows the equivalence class vectors of all sofas in the database.

Object	Equivalence classes	
Sofa_Alatea	COLOUR	II
	MATERIAL	I
Sofa_Consuelo	COLOUR	III
	MATERIAL	I
Sofa_Grace	COLOUR	II
	MATERIAL	I
Sofa_Nadia	COLOUR	III
	MATERIAL	0
Sofa_Isadora	COLOUR	I
	MATERIAL	I

Table 4: Catalogue items with their respective equivalence classes.

The distribution of sofas in the resulting two-dimensional vector space can be seen in Figure 1. Finding the set K of optimal candidates with respect to the users preferences is now a purely geometrical problem as stated in Theorem 1.

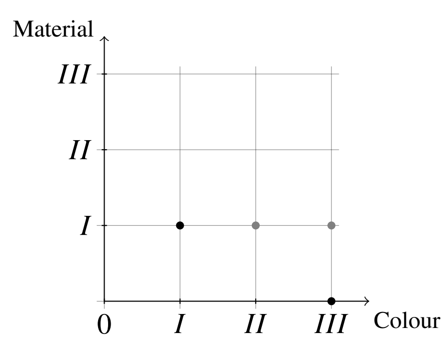

Figure 1: Geometric representation of the search space for optimal candidates.

The only elements of K are the points (III,0) and (I,I). If we compare the respective sofas to sofas assigned to other points, we can see that there exists at least one dimension in which the elements of K are better. Each point in K corresponds to a cube in the corresponding conceptual space defined by the properties material and colour. Each of the two optimal cubes contains exactly one sofa. Hence, we end up with sofa Nadia and sofa Isadora. Their values for colour and material can now be used to generate an answer which informs the customer about the best available alternatives. First, the customer must be informed that there is no object in the database which meets his preferences directly, e.g. by producing '*I'm afraid we don't have a purple leather sofa*'. Then, he has to be informed about the optimal alternatives. For this step, we first consider the feature structures of the two sofas as specified in the catalogue, see Table 5. The problem which has to be solved now

$$\begin{bmatrix} \text{NAME} & \text{Sofa\_Nadia} \\ \text{COLOUR} & \text{black} \\ \text{MATERIAL} & \text{leather} \end{bmatrix} \quad \begin{bmatrix} \text{NAME} & \text{Sofa\_Isadora} \\ \text{COLOUR} & \text{amethyst} \\ \text{MATERIAL} & \text{fabric} \end{bmatrix}$$

Table 5: Sofa Nadia and Isadora

is the verbalisation of this set of alternatives. For example, the customer may not know which colour *amethyst* is, in particular, he may not know that it is a shade of purple. We therefore restricted the colour terms which may occur in answers to basic colour terms, which the customer can be assumed to know. The next basic colour term which is higher in the colour ontology than *amethyst* is *purple*. The colour of Sofa Nadia *black* already is a basic colour term. For material, the catalogue only contains basic properties which are commonly known. Hence, we can generate the sentence '*I can show you a purple fabric sofa or a black leather sofa*'. Adding '*but*' to mark contrast we arrive at the answer given in (3), repeated here as (4):

(4) USR: I would like to have a *purple leather* sofa.
NPC: I'm afraid we don't have a purple leather sofa, but I can show you a *purple fabric* sofa or a *black leather* sofa.

In the introduction, we said that a response as in (3) is not only conveying literal information about available alternatives but also the implicature that to the speaker's best knowledge there are no alternatives which are better than those mentioned. This can now be put more precisely as meaning that there are no property combinations which would be closer to the target than the combinations mentioned as alternatives. This condition is automatically satisfied by the construction of the answer. If the answer is understood to be exhaustive, then it even follows that the reamining alternatives are worse than those presented. In order to make the answer exhaustive, all elements of the efficient frontier K had to be presented. This goal can only be met if the size of K does not contain more than three to four elements. The model therefore predicts that only small numbers of alternatives, i.e. one or two, are interpreted as exhaustive, and that more answers with three or four alternative property combinations are ambiguous between being exhaustive and not exhaustive. It remains to be tested whether humans in their conversation make the exhaustive interpretation dependent on the number of alternatives. If the answer does not mention all elements of K, then it may be that the customer would prefer one of the unmentioned elements. But even in this case, the implicature that there is no closer alternative property combination than the mentioned one is true.

Finally, we want to motivate our division of the different dimensions into intervals and, thereby, the division of the search space into cubes of roughly equivalent property-value combinations. Instead, we could have directly searched for a list of Pareto efficient database objects. The first reason for our approach is that the division of the dimensions into intervals results in a coarser-grained search space, and, consequently, in a smaller K. Second, as Example (3) shows, the sales agent's answer proposes alternative property-value combinations, not objects, each of which denotes an alternative area in the conceptual space. Hence, the goal of our search is, at this stage, to find property-value combinations which can be presented verbally. Third, that K contains all possibly optimal alternatives depends on a number of assumptions, one of them being the assumption that the user's utility function F strictly increases with distance from the target; another one being the assumption that all relevant attributes are known. Especially the later will not be met in practice. A small difference of the colour shade of two objects will not necessarily outweigh all other differences with respect to unnamed attributes, such as e.g. shape, size, style, or price. By dividing each dimension into a set of intervals we make sure that the differences between the database objects falling in different intervals are large enough so that the preferences for them are also significantly different. Finally, each cube is representative of a different trade-off. Presenting cubes, thus, already guarantees diversity in the presentation set.

4 Related work

In the recommender systems literature we find many approaches to the generation of cooperative responses to over-constrained queries. Most of them consider only situations in which the weights of the different preferences are known. A common approach is to propose the user one or several query relaxations. A *query relaxation* means that some constraints expressing user preferences are dropped so that the remaining constraints can be satisfied by some catalogue object. Query relaxations are usually computed on the basis of a ranking of attributes such that weaker constraints are proposed for relaxation first, e.g. [9, 13, 14, 18]. Additional criteria may be considered such as e.g. the minimality of the subset of constraints chosen for relaxation [9, 13], or the density of the constraints measured as the amount of items in the search result [9, 14]. Such approaches suffer in general from the problem that the extent to which a constraint must be violated is not taken into account. To illustrate this point, consider the situation in which the user has requested a "lilac wallpaper with floral pattern" and the available options are "lilac wallpaper with stripes" and "pink wallpaper with floral pattern". Even if the user has a strong preference for colour over pattern, the second option is still interesting for him, since colour is violated only to a small degree, while in the first option pattern suffers from a more dramatic violation. These approaches would only select or rank higher the option preserving the colour. A similar situation arises when a query relaxation involves a small violation of more than one constraint and a second query relaxation involves a strong violation of a single constraint. The second option would be preferred by most of those approaches.

This problem is overcome by *decision-theoretic* approaches to item retrieval, such as [6, 12, 19], among others. In these approaches items are ranked according to overall similarity to the requested item, where overall similarity is computed as the weighted sum of local similarity measures for the specified attributes, as shown by the following equation[7]:

$$F(e) = \sum_{i=1}^{n} \alpha_i F_i(e_i) \tag{4.3}$$

where e is an item, N the number of attributes, e_i and α_i the value and the weight of attribute i, respectively, and F a utility function. McSherry [12] and White et al. [19] do not only include the item with the highest score in the retrieval set, but also those items with the highest score that represent each a different possible trade-off, ensuring, thus, diversity in the retrieval set[8].

Our approach is in line with these decision-theoretic approaches. Our main contribution with respect to them is the assessment of preference of one alternative over another based on similarity. In general, these approaches do not consider how the similarity measures are obtained or represented. They do not assume any model of concept representation. In our work, by representing the search space as conceptual spaces, we explicitly focus on the preference assessment part of the task.

Another difference with these approaches is that they assume that the strengths of the different preferences are known to the system, while we consider the situation in which the preferences are of qualitative nature. Faltings *et al.* [7] do also consider the situation in which the weights of the different preferences are unknown. They discuss three qualitative models of preferences: a dominance-based one which retrieves all Pareto-optimal candidates (undominated candidates), a utilitarian one which minimizes overall penalty and an egalitarian one which minimizes maximal penalty. While the dominance-based model does not make any assumption about the preferences and retrieves all possible trade-offs, the utilitarian model assumes that overall similarity to the requested item, that is, overall smaller violations are preferred, while the egalitarian model assumes that strong violations are dispreferred. With an increasing number of preferences, Faltings *et al.* [7] conclude that the utilitarian and egalitarian filters are superior to the dominance-based one. However, according to their results, the probability for the dominance-based filter of retrieving all Pareto-optimal

[7] McSherry [12] uses a variant of this formula that additionally divides the sum through the sum of the weights for the different attributes.

[8] Diversity in retrieval sets has been an important topic of recent research in the area of recommender systems.

items for a small number of preferences and retrieval sizes that range from .046 to 7 % of the whole catalogue is quite high (e.g. 100% for one and two preferences, around 68% for three preferences). This makes the dominance-based filter suitable for our scenario, where at most four preferences are stated, but mostly just two or three, and the allowed retrieval set sizes are within the limits considered by Faltings *et al.* [7][9]. In next section, we present additional filtering mechanisms which allow us to reduce the retrieval set in cases in which it becomes too large for verbal presentation.

5 Filtering and ranking the retrieved alternatives

In section 3 we explained that the division of the dimensions in the search space into intervals already guarantees a smaller retrieval set. However, summarizing options in cubes involves sacrificing accuracy, that is, cubes could contain dominated alternatives. There are two solutions for this problem. One possibility is to have cubes of different sizes: smaller cubes for shorter distances and increasingly larger cubes for larger distances. The grouping of distances in intervals of different sizes is in consonance with the idea that *perceived similarity* [17] exponentially decays with increasing distance to the target. Although perceived similarity is measured as the probability that two stimuli obtain the same response, we can generalize it to our task, by assuming that from a certain distance objects are (almost) equally unacceptable for the user. This allows us to preserve accuracy for short distances, while keeping the amount of cubes small. Another possibility simply involves filtering dominated items within a cube out according to the original local similarity measures.

Finally, in the furniture sales scenario domain knowledge supports the selection of a subset of Pareto efficient elements without arbitrarily weighting the attributes. In Example 1 we have seen that similarity to already existing furniture plays an important role. Preference statements as e.g. '*sideboard similar to my shelf*' can be treated in the same way as preference statements of the form '*a white sideboard*'. The only difference is that the target t is not defined by explicitly stated properties but by the properties of the object of comparison. In general, constraints which state that the searched piece of furniture must *harmonize* with existing furniture can be added by default. This means that a selection from K can be made on the basis of a function which measures how well new objects x harmonize with existing objects t. This means, if $t_1, \ldots, t_m$ are the relevant objects with which the new object should harmonize, then we can rank the objects in the cubes $e \in K$ according to the $\min\{d(x_e, t_1), \ldots, d(x_e, t_m)\}$, and select the property combinations of the best objects in the three or four best cubes for presentation. The ranking also provides us with an order for showing objects once a particular property combination has been chosen.

6 Conclusion

We have presented an approach to finding optimal alternative search space areas to serve as the basis for the generation of optimal cooperative responses to overconstrained queries. Our approach computes the complete set of optimal alternatives without assuming any particular weights for the different attributes. Our main contribution is the connection of Decision Theory with a cognitive model of concept representation which allows, based on a natural similarity measure, to constrain the values of utility functions. Several methods have been proposed for non-arbitrarily reducing the size of the retrieval set.

The solution proposed is not only valid in a situation in which no items meet the requirements imposed by the user, but also in a situation in which all items meeting the requirements have been shown and plainly rejected by the user. In such a situation, the system also has to come up with further alternatives to propose. The approach presented in this paper can be applied in this situation without modification, provided only that the rejected items are excluded from the search space.

For the dialogue capabilities of the NPC to be human-like, this has not only to convey correct literal information but also make sure that the implicatures that a human addressee will automatically infer from the answer hold true. For example, the human addressee will infer that, to the speaker's best knowledge, the alternatives are among the best he can offer. This implicature is automatically satisfied by the construction of the answer.

[9] We set our retrieval set size to four, which corresponds to what is generally assumed to be the upper limit on the amount of items which can be verbally presented without imposing too much cognitive load on the user. With a catalogue of up to 869 items we will still be within the relative retrieval set sizes considered by these authors in their experiment.

Currently, we are working on the content planning component of the answer generation. In order to guarantee that the customer finds the object that best matches his preferences, an optimal global strategy involves reducing as much uncertainty as possible regarding the acceptability of the different options (especially the interesting ones) in the shortest possible dialogue. If the system has information that an option is much more preferred than the others, it will proceed to show an object representative of that option. If, otherwise, there is no such evidence, the system will have to find out how the user stands to the available alternatives and then present objects accordingly. Often, only a subset of the alternatives can be presented. Our approach represents the system's beliefs about the acceptability of the different options in a probabilistic network. The system will choose the alternatives so, that they represent as many trade-offs as possible and that finding out how the user stands to them allows to draw more inferences about the acceptability of the different items.

References

[1] Aisbett, J. and Gibbon, G. (2001). A general formulation of conceptual spaces as a meso level representation. *Artificial Intelligence*, 133(1-2):189–232.

[2] Asher, N., Bonzon, E., and Lascarides, A. (2010). Extracting and modelling preferences from dialogue. In Hüllermeier, E., Kruse, R., and Hoffmann, F., editors, *Computational Intelligence for Knowledge-Based Systems Design*, volume 6178 of *Lecture Notes in Computer Science*, pages 542–553. Springer.

[3] Bertomeu, N. and Benz, A. (2009). Annotation of joint projects and information states in human-NPC dialogue. In *Proceedings of the First International Conference on Corpus Linguistics CILC-09*, pages 723–740.

[4] Boutillier, C., Brafman, R. I., Domshlak, C., Hoos, H. H., and Poole, D. (2004). CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21:135–191.

[5] Boutillier, C., Brafman, R. I., Hoos, H. H., and Poole, D. (1999). Reasoning with conditional ceteris paribus preference statements. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence UAI-99*, pages 71–80.

[6] Carberry, S., Chu-Carroll, J., and Elzer, S. (1999). Constructing and utilizing a model of user preferences in collaborative consultation dialogues. *Computational Intelligence*, 15:185–217.

[7] Faltings, B., Torrens, M., and Pu, P. (2004). Solution generation with qualitative models of preferences. *In Computational Intelligence*, 20:246–263.

[8] Gärdenfors, P. (2000). *Conceptual Spaces: The Geometry of Thought*. The MIT Press, Cambridge, MA.

[9] Jannach, D. (2009). Fast computation of query relaxations for knowledge-based recommenders. *AI Communications*, 22:235–248.

[10] Keeney, R. L. and Raiffa, H. (1993). *Decisions with Multiple Objectives - Preferences and Value Tradeoffs*. Cambridge University Press, Cambridge.

[11] McNamara, P. (2010). Deontic logic. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Fall 2010 edition.

[12] McSherry, D. (2003). Similarity and compromise. In *In Proceedings of the Fifth International Conference on Case-Based Reasoning*, pages 291–305. Springer.

[13] McSherry, D. (2005). Retrieval failure and recovery in recommender systems. *Artificial Intelligence Review*, 24:319–338.

[14] Qu, Y. and Beale, S. (2002). Cooperative resolution of over-constrained information requests. *Constraints*, 7:29–47.

[15] Raubal, M. (2004). Formalizing conceptual spaces. In *Formal Ontology in Information Systems, Proceedings of the Third International Conference (FOIS 2004)*, volume 114, pages 153–164, Amsterdam. NL: IOS Press.

[16] Ross, A. (1941). Imperatives and logic. *Theoria*, 7:53–71.

[17] Shepard, R. N. (1987). Toward a universal law of generalization in psychological science. *Science, New Series*, 237(4820):1317–1323.

[18] Thompson, C. A., Göker, M. H., and Langley, P. (2002). A personalized system for conversational recommendations. *Journal of Artificial Intelligence Research*, 21:393–428.

[19] White, M., Clark, R. A. J., and Moore, J. D. (2010). Generating tailored, comparative descriptions with contextually appropriate intonation. *Computational Linguistics*, 36(2):159–201.