Tailoring Evaluative Arguments to User's Preferences

Giuseppe Carenini[1] and Johanna Moore[2*]

[1] Intelligent Systems Program, University of Pittsburgh, U.S.A.
[2] HCRC, University of Edinburgh, U.K.

Abstract. Computer systems that serve as personal assistants, advisors, or sales assistants frequently need to argue evaluations of domain entities. Argumentation theory shows that to argue an evaluation convincingly requires to base the evaluation on the hearer's values and preferences. In this paper we propose a framework for tailoring an evaluative argument about an entity when user's preferences are modeled by an additive multiattribute value function. Since we adopt and extend previous work on explaining decision-theoretic advice as well as previous work in computational linguistics on generating natural language arguments, our framework is both formally and linguistically sound.

1 Introduction

Computer systems that serve as personal assistants, advisors, or sales assistants frequently need to generate evaluative arguments for domain entities. For instance, a student advisor may need to argue that a certain course would be an excellent choice for a particular student, or a real-estate personal assistant may need to argue that a certain house would be a questionable choice for its current user. Argumentation theory indicates that to argue an evaluation convincingly requires to base the evaluation on the hearer's values and preferences. Therefore, the effectiveness of systems that serve as assistants or advisors in situations in which they need to present evaluative arguments critically depends on their ability to tailor their arguments to a model of the user's values and preferences. In this paper we propose a computational framework for generating evaluative arguments that could be applied in systems serving as personal assistants or advisors. In our framework, as suggested by argumentation theory, arguments are tailored to a model of user's preferences. Furthermore, in accordance with current research in user modeling, we adopt as model of user's preferences a conceptualization based on multiattribute utility theory (more specifically an additive multiattribute value function (AMVF)).

2 Background on AMVF

An AMVF is a utility model based on a *value tree* and on a set of *component value functions*, one for each attribute of an alternative. A value tree is a decomposition of the objective to maximize the value of the selection for the decision maker into an objective hierarchy in which the leaves correspond to attributes of the alternatives. The arcs of the tree are weighted depending on the

* This work was supported by grant number DAA-1593K0005 from the Advanced Research Projects Agency (ARPA). Its contents are solely the responsibility of the authors and do not necessarily represent the official views of either ARPA or the U.S. Government.

importance of an objective in achieving the objective above it in the tree. Note that the sum of the weights at each level is equal to 1. A component value function for an attribute expresses the preferability of each attribute value as a number in the $[0, 1]$ interval. Formally, an AMVF has the following form:

$$v(a) = v(x_1, ..., x_n) = \sum_{i=1}^{n} w_i v_i(x_i) \tag{1}$$

- $(x_1, ..., x_n)$ is the vector of attribute values for alternative a
- For each attribute i, v_i is the *component value function*, which maps the least preferable x_i to 0, the best x_i to 1, and the other x_i to values in [0,1]
- w_i is the *weight* for attribute i, and $0 < w_i < 1$ and $\sum_{i=1}^{n} w_i = 1$

3 The Framework and a Sample Argumentative Strategy

Our framework for generating evaluative arguments is based on previous work in artificial intelligence on explaining decision-theoretic advice (Klein and Shortliffe, 1994), and on previous work in computational linguistics on generating natural language evaluative arguments (Elhadad, 1995) . On the one hand, the study on explaining decision-theoretic advice produced a rich quantitative model that can serve as a basis for strategies to select and organize the content of decision-theoretic explanations, but was not concerned at all with linguistic issues. On the other hand, the work in computational linguistics produced a well-founded model of how argumentative intents (i.e., whether a proposition favors or disfavors the alternative) can influence sophisticated linguistic decisions, but it produced weak results as far as the selection and the organization of the argument content is concerned.

From Klein and Shortliffe (1994) we adopted and adapted two basic concepts for content selection and organization: $\mathtt{s-compellingness}$ and $\mathtt{notably-compelling}?$. An objective can be compelling in arguing for an alternative either because of its strength or because of its weakness in contributing to the value of an alternative. So, if x measures how much the value of an objective is contributing to the overall value difference of an alternative from the worst case[1] and y measures how much the value of an objective is determining the overall value difference of an alternative from the best case, a possible definition for $\mathtt{s-compellingness}$ is the greatest of the two quantities x and y. Informally, an objective is $\mathtt{s-notably-compelling}?$ if it is an outlier in a population with respect to $\mathtt{s-compellingness}$.

Elhadad's techniques for performing sophisticated linguistic decisions based on argumentative intent used a model of user's preferences different from an AMVF. So, in order to adopt his work, we have to show how argumentative intents can be computed from an AMVF-based model of user's preferences. Basically, two subintervals of the interval $[0, 1]$, $[0, k_-]$ and $[k_+, 1]$ must be defined (e.g., $[0, 0.4]$ and $[0.6, 1]$). Then we consider the value of an alternative for an objective to correspond to a negative, positive or neutral argumentative intent depending on whether it belongs to $[0, k_-]$, $[k_+, 1]$ or $[k_-, k_+]$ respectively.

We sketch now a sample argumentative strategy for the communicative goal of evaluating a single entity. The strategy is based on the notions of $\mathtt{s-compellingness}$ and $\mathtt{notably-compelling}?$ defined previously. First, we introduce some terminology: *Root* is the objective the argument is

[1] a_{worst} is an alternative such as that $\forall o \; v_o(a_{worst}) = 0$, whereas a_{best} is such as that $\forall o \; v_o(a_{best}) = 1$.

$User = user-b, Entity = house-b15, k = -1, k_- = 0.4, k_+ = 0.6, v(house-b15, user-b) = 0.741$

```
I would strongly suggest house-b15. It is spacious. It has a large lot-size (800 sqf),
many rooms (7) and plenty of storage space (200 sqf). Furthermore, since it offers an
exceptional view and appears nice, house-b15 quality is excellent. Finally, house-b15 has a
fair location. Although it is far from a shopping area, it has access to a park (2 miles).
```

Figure 1. Sample concise and fluent natural language evaluative argument

about; *MainArgIntent* is either + or - and determines whether the generated argument will favor or disfavor the entity; *ArgItent* is a function that when applied to the value of an objective returns its argumentative intent (which is either +, - or neutral); the *Express* function indicates that an objective must be realized in natural language with a certain argumentative intent.

Argumentation Strategy

argue(Root, entity, user-model, MainArgIntent, k, k_-, k_+)

;; content selection and assignments

Eliminate all objectives o_i for which $s-notably-compelling?(o_i, siblings(o_i), a, Root)$ is false.

$MainFactorInFavor \leftarrow$ most compelling objective o such as that $o \in children(Root)$ and $ArgItent(v_o(entity)) = MainArgIntent$

$ContrastingSubObjectives \leftarrow$ all o_i such as that $o_i \in children(Root)$ and $ArgIntent(v_{o_i}(entity)) \neq MainArgIntent$

$RestInFavor \leftarrow$ all o_i such as that $o_i \in children(Root)$ and $o_i \neq MainFactorInFavor$ and $ArgItent(v_{o_i}(entity)) = MainArgIntent$

;; Steps for expressing the content.

Express(Root, MainArgIntent) ;; (e.g., "I strongly suggest this entity")

Argue($MainFactorInFavor$, entity, user-model, MainArgIntent, k, k_-, k_+)

For all $o_i \in ContrastingSubObjectives$, Express($o_i$, $ArgIntent(v_{o_i}(entity))$) ;;ordered by compellingness

For all $o_i \in RestInFavor$, Argue(o_i, entity, user-model, MainArgIntent, k, k_-, k_+) ;;ordered by compellingness

This strategy is based on guidelines for presenting evaluative arguments suggested in argumentation literature (Mayberry and Golden, 1996). The main factor in favor of the evaluation is presented in detail, along with possible counter-arguments (i.e., $ContrastingSubObjectives$), that must always be considered, but not in detail. Finally, further supporting factors must be presented in detail. As specified in the strategy, details about a factor are presented by recursively calling the function $argue$ on the factor. Figure 1 shows a sample argument generated by the strategy when it is applied to a user model based on a value tree consisting of 15 objectives.

As future work, we plan to empirically evaluate our framework. We also intend to extend it to more complex models of user preferences and to generating arguments that combine language with information graphics.

References

Elhadad, M. (1995). Using argumentation in text generation. *Journal of Pragmatics* 24:189–220.

Klein, D. A., and Shortliffe, E. H. (1994). A framework for explaining decision-theoretic advice. *Artificial Intelligence* 67:201–243.

Mayberry, K. J., and Golden, R. E. (1996). *For Argument's Sake: A Guide to Writing Effective Arguments*. Harper-Collins, College Publishers. Second Edition.