Discernability and Preference in Interactive Option Searches

Michael Minock
Department of Computing Science
Ume University, Sweden 90187
mjm@cs.umu.se

Abstract

In option searches, a user seeks to locate an ideal option (e.g. a flight, restaurant, book, etc.) from a set of n such options. The aim of this paper is to provide a solid mathematical basis for optimizing presentation length in such searches. The paper develops an information theoretic model that takes into account the user's ability to discern among options as well as their *a priori* preference. The developed model makes definite predictions about what clusterings of a user query are more or less informative based on measures of *information gain.* Users are offered descriptions of such clusters as the basis for subsequent refinement steps in a drill-down dialogue to locate the best option. We have implemented an initial system that performs reasonably well on moderately large data sets and gives intuitively appealing results. The system is in the process of being integrated into a natural language interface system for end-user evaluation.

1 Introduction

As pointed out in [5], it is critical that spoken dialogue systems limit presentation duration for interactive option searches. Thus if the user requests "flights to Berlin leaving before noon," and there are many such flights, it is a mistake to simply start listing them in succession – the user would become irritated by the long descriptions and would be unlikely to remember enough detail to make an optimal choice. In the database of table 1 there are just four such flights, but even here it might be better to ask the follow up question, "Do you prefer Lufthansa or SAS?". Such *summarize-and-refine* (SR) techniques [4] cluster the options meeting the user's constraints into sets (e.g. "the SAS flights" and "the Lufthansa flights"), present these sets as summaries (or implicitly through questions) and then let the user refine the search to the cluster that interests them most. Such techniques promote efficiency by reducing what would be a linear number of descriptions to a roughly logarithmic number.

While such *summarize-and-refine* systems are particularly suited to spoken dialogue systems where users can reliably command systems to drill down into one or another summary, there are difficulties when such systems pick summaries that are not discernible to the user. For example if the system were to respond to the question above with "do you prefer flights on an A300 or an A320?", most users would be hard pressed to make an informed choice. The work presented here, inspired by [1], recasts the interactive search process in an information theoretic light and introduces a model of *discernibility* among options as well as a general parameter γ of intolerance for a sub-optimal results. The work's main contribution is to propose a more solid mathematical basis for optimizing presentation length in options searches.

2 Foundations

2.1 Options, databases, answer sets and clusterings

Consider the universe of values $\mathcal{U}$ and, for a given k, all the k-tuples $\mathcal{U}^k$, hereafter referred to as *options.* We denote the i-th value (starting at 1) of option t as $t[i]$. The set of *conditions* $\mathcal{C}$ are functions mapping $\mathcal{U}^k \rightarrow \{\mathsf{true}, \mathsf{false}\}$, that is for $c \in \mathcal{C}$ and option $t \in \mathcal{U}^k$, $c(t)$ is either true or false. Let $\mathcal{D}$ be a database of n options $t_1, .., t_n$. An *answer*

no.	dest	airline	dep	price	meal	aircraft
1	Paris	SAS	8	€200	yes	A300
2	Berlin	Luft	8	€250	yes	A320
3	London	SAS	9	€150	yes	A300
4	Paris	AF	9	€250	yes	A320
5	Berlin	Luft	9	€200	no	A320
6	London	BA	10	€200	yes	A320
7	Berlin	SAS	10	€250	no	A300
8	Berlin	SAS	11	€100	no	A300

Table 1: Example Last Minute Travel Database

set is denoted as $\{x | x \in \mathcal{D} \wedge Q(x)\}$ where $Q(x)$ is a boolean combination of conditions. Hereafter we will assume that $\mathcal{D}$ is fixed and thus drop explicit reference to it, instead describing answers sets as simply $\{x|Q(x)\}$. The semantics of answer sets are standard, where $(\forall t \in \mathcal{D})(t \in \{x|Q(x)\} \Leftrightarrow Q(t))$. Often we will refer to the expression $Q(x)$ as a *query*.

When deciding the next dialogue move after the user has identified $\{x|Q(x)\}$ as the set that they interested in, we must consider the possible clusterings $\langle Q(x) : Q_1(x), .., Q_m(x)\rangle$ which present m further summarize-and-refine sets to consider. As an example, the clustering of the query for "the flights to Berlin leaving before noon" into those on Lufthansa or SAS is:

$$\begin{aligned}&\langle\{x|\mathsf{beforeNoon}(x) \wedge \mathsf{toBerlin}(x)\} :\\ &\quad \{x|\mathsf{beforeNoon}(x) \wedge \mathsf{toBerlin}(x) \wedge \mathsf{onLuft}(x)\},\\ &\quad \{x|\mathsf{beforeNoon}(x) \wedge \mathsf{toBerlin}(\mathsf{x}) \wedge \mathsf{onSAS}(x)\}\rangle\end{aligned}$$

Note that our definition of a clustering puts no conditions on the relationship between $\{x|Q(x)\}$ and $\cup_{i=1}^{m}\{x|Q_i(x)\}$. Thus the relationship may be specialization, generalization or some combination thereof. For example a 'specializing' clustering of "the flights to Berlin leaving before noon" into those €100 euro or less is:

$$\begin{aligned}&\langle\{x|\mathsf{beforeNoon}(x) \wedge \mathsf{toBerlin}(x)\} :\\ &\quad \{x|\mathsf{beforeNoon}(x) \wedge \mathsf{toBerlin}(x) \wedge\\ &\quad\quad \mathsf{PriceLEQ}(x, 100)\}\rangle\end{aligned}$$

A 'generalizing' clustering could be:

$$\begin{aligned}&\langle\{x|\mathsf{beforeNoon}(x) \wedge \mathsf{toBerlin}(x)\} :\\ &\quad \{x|\mathsf{before3PM}(x) \wedge \mathsf{toBerlin}(x) \wedge \mathsf{onLuft}(x)\},\\ &\quad \{x|\mathsf{before1PM}(x) \wedge \mathsf{toBerlin}(\mathsf{x}) \wedge \mathsf{onSAS}(x)\}\rangle\end{aligned}$$

2.2 User preferences

A model of user preference captures the *a priori* assumptions about how the user values alternative options. Note that within a specific dialogue, users express hard conditions such as the destination or the need to fly at a specific time that are not captured in the user model. However given that a set of options meet the hard constraints supplied by the user, the user model will rank these options based on this *a priori* model. Moreover as we shall see below, based on notions of discernibility, options that fall outside of the user supplied hard constraints may in fact be worth presenting.

The work here allows for any type of quantitative model of user preference, but to avoid formal difficulties, we assume that for all $t \in \mathcal{D}$, $\mathsf{util}(t) > 0$. The following shorthand notation expresses total utility over an answer set:

$$\mathsf{util}(\{x|Q(x)\}) = \sum_{t \in \{x|Q(x)\}} \mathsf{util}(t)$$

2.3 Discernibility

In addition to the model of preference, there is a related model of discernibility. Two options are perfectly discernible if the user can immediately recognize them as being qualitatively different. For example, under a 'normal' context, a flight to Berlin versus a flight to Paris are perfectly discernible where as two flights to Berlin, one on an A300 and other on an A320 are not discernible.

Formally, for each i-th component of the options, assume that there is a function $\zeta_i : \mathcal{U} \times \mathcal{U} \rightarrow [0..1]$. The intuition of ζ_i is that if options t and t' agree on all components other than i (i.e. $t[j] = t'[j]$ for $1 \leq j \leq k$ and $j \neq i$), then $\zeta_i(t[i], t'[i])$ is the probability that t and t' are indistinguishable to the user. The product of these measures gives an overall measure of similarity for tuples.

$$\mathsf{sim}(t, t') = \prod_{i=1}^{k} \zeta_i(t[i], t'[i])$$

Note that $\mathsf{sim}(t,t) = 1$ and that $\mathsf{sim}(t,t') = 0$ if there is at least one component upon which t and t' are perfectly discernible.

3 Our Approach

3.1 The ideal answer assumption

We make what we call the **ideal answer assumption** which states that there is some option $\mathsf{opt} \in \mathcal{D}$ which is the single best option that the user is searching for. The amount of information that can be usefully applied to locating opt is measured in bits, or answers to 'yes/no' questions. While in cases of perfect discernibility, it will take $\log_2 n$ bits to locate opt among n options, due to problems of discernibility only so many bits may be usefully employed to locate opt. Note that this is different from a measure of entropy. Consider the cases in which all options are completely indiscernible. Answering yes/no questions provides no information toward locating the ideal option. The best one can do in fact is simply pick one the options at random and present it as the ideal. Formally, we use the following definition of the information content within a cluster:

$$I(\{x|Q(x)\}) = \log_2\left(\frac{|\{x|Q(x)\}|^2}{\sum_{t' \in \{x|Q(x)\}} \sum_{\hat{t} \in \{x|Q(x)\}} \mathsf{sim}(t', \hat{t})}\right)$$

The prior probability of an option being ideal is proportional to its utility with respect to the model of *a priori* user preferences:

$$P(t = \mathsf{opt}) = \frac{\mathsf{util}(t)}{\mathsf{util}(\{x|x \in \mathcal{D}\})}$$

We introduce the notation $\mathsf{id}_{\hat{t}}$ to denote the situation in which the user has identified the option $\hat{t}$ as opt. Of course, based on problems of discernibility, the user could be wrong.

$$P(t = \mathsf{opt}|\mathsf{id}_{\hat{t}}) = \frac{\mathsf{sim}(t, \hat{t})}{\sum_{t' \in \{x|x \in \mathcal{D}\}} \mathsf{sim}(t', \hat{t})}$$

We now introduce the generalized notation id_Q to denote the situation in which the user has declared that $\mathsf{opt} \in \{x|Q(x)\}$. We obtain:

$$P(t = \mathsf{opt}|\mathsf{id}_Q) = \sum_{\hat{t} \in \{x|Q(x)\}} P(t = \mathsf{opt}|\mathsf{id}_{\hat{t}}) \cdot \frac{\mathsf{util}(\hat{t})}{\mathsf{util}(\{x|Q(x)\})}$$

Note that we are weighing options in $\{x|Q(x)\}$ according to the model of user preference. This makes sense, because the model of preference gives us our *a priori* probability that a given option within $\{x|Q(x)\}$ would be selected as ideal by the user. Now we develop the full generalized form:

$$P(\mathsf{opt} \in \{x|Q'(x)\}|\mathsf{id}_Q) = \sum_{t' \in \{x|Q'(x)\}} P(t' = \mathsf{opt}|\mathsf{id}_Q)$$

3.2 Information Gain

The natural question to consider is how much information is gained through a clustering $\langle Q(x) : Q_1(x), .., Q_m(x)\rangle$. This is decided in the normal way by subtracting the information required to locate opt within $\{x|Q(x)\}$ from the information required to locate opt within each cluster $\{x|Q_i(x)\}$ weighted by the probability that the user will select the given cluster $Q_i(x)$ on their next refinement move. Finally consideration must be given to the possibility that the user refines the wrong cluster or that opt is not within any cluster $\{x|Q_i(x)\}$. In such a case the user suffers the cost γ measured in terms of bits. Given these ideas we arrive at the following measure of gain:

$$\begin{aligned}
&\mathsf{gain}(\langle Q(x) : Q_1(x), .., Q_m(x)\rangle) = \\
&\quad I(\{x|Q(x)\}) + P(\mathsf{opt} \notin \{x|Q(x)\}|\mathsf{id}_Q) \cdot \gamma \\
&\quad - \\
&\quad \sum_{i=1}^{m} P(\mathsf{sel}_{Q_i}) \cdot (I(\{x|Q_i(x)\}) + \\
&\qquad\qquad P(\mathsf{opt} \notin \{x|Q_i(x)\}|\mathsf{id}_{Q_i}) \cdot \gamma) \\
&\quad -P(\mathsf{opt} \in \{x|Q(x) \bigwedge_{i=1}^{m} \neg Q_i(x)\}|\mathsf{id}_Q) \cdot \gamma
\end{aligned}$$

where sel_{Q_i} means that the user will select Q_i as the basic of further refinement.

Using the model of user preferences we assume[1] that:

$$P(\mathsf{sel}_{Q_i}) = \frac{\mathsf{util}(\{x|Q_i(x)\})}{\sum_{Q' \in \{Q_1,..,Q_m\}} \mathsf{util}(\{x|Q'(x)\})}$$

That is to say that the probability of a user selecting a set corresponds to the total utility within the set relative to the total utility of all sets under consideration.

3.3 Decision procedure

Given a non-empty $Q(x)$, we generate a set of alternative clusterings, picking the one with *highest benefit*. Benefit is determined by the dividing information gain by the *cost* of summarizing the clusters to the user. Formally we pick $\hat{s}$ in:

$$\hat{s} = \arg\max_{s \in S}\left(\frac{\mathsf{gain}(\langle Q(x) : Q_1(x), ..., Q_m(x)\rangle)}{\mathsf{cost}(s)}\right)$$

where $s = \langle Q(x) : Q_1(x), .., Q_m(x)\rangle$ and S is the set of clustering statements. To keep things simple we assume that the cost of reporting the clustering $\langle Q(x) : Q_1(x), .., Q_m(x)\rangle$ is simply m. This assumption is of course too simplistic – a more reasonable measure, though beyond the scope of this paper, would be based on the cost of presenting the clustering in natural language.

Because S is (practically) infinite, we must give up on optimality and instead generate a representative sample $S' \subset S$. This set of clusterings is built randomly through splitting $Q(x)$ via new conditions and then by specializing (or generalizing) or further splitting the resulting clusters. Our methods to calculate gain are purely distributional. That is we directly compute our measures through iterating over answers sets yielded by our clusters. The calculation of gain is $O(m \cdot n^3)$ for the n options under consideration and a clustering of m clusters. Thus if we recast our problem as a search problem, the evaluation function is polynomial in the size of the problem. Although our current method to obtaining S' is still rather naive, such methods can achieve reasonable performance for moderately sized data sets.

[1]There are several other reasonable models that can be used here. For example the average utility or even a more complex measure of perceived utility based on discernibility.

Thus far we have left the set of conditions $\mathcal{C}$ unspecified. The conditions are just boolean mappings over options (or k-tuples). While the condition $\mathsf{PriceIsPrime}(x)$ may return true for all options where the fifth component is a prime number, there are an infinite number of such far fetched conditions and thus we isolate attention to a fixed finite set of conditions $\mathcal{C}_{\mathsf{simple}} \subseteq \mathcal{C}$ which are the conditions that 'make sense' in the given domain.

Given $\mathcal{C}_{\mathsf{simple}}$, the set $\mathcal{Q}$ of semantically distinct queries that may be built up as boolean formulas from conditions within $\mathcal{C}_{\mathsf{simple}}$. Note that $\mathcal{Q}$ is large, though finite. We assume here that the natural language interface may relate user typed (or spoken) strings to elements within $\mathcal{Q}$ for the purposes of understanding and paraphrasing.

4 Example

Although we have a working demonstration system, we choose here to present a series of examples to illustrate the properties of our algorithm over the database of table 1.

4.1 Three user models and a model of discernibility

To simplify the presentation we assume a very simple linear user model based on the coefficients a_i and b_i. To achieve this we capture a value mapping function v_i for the i-th option component values to numerical measures: $v_i(\mathcal{U}) \rightarrow \mathbb{R}$. Assume that $v_i(z) = z$ for numerical values and $v_i(z) = 1$ for non-numerical values. The default utility of an option is thus measured as:

$$\mathsf{util}(t) = \sum_{i=1}^{k} a_i \cdot v_i(t[i]) + b_i$$

We present three user models. The first is for Maxwell Entropy: $a_i = 0, b_i = \frac{1}{7}$. As we can see, Max has no default preference for one option over another. The second user model is that of the student who only favors one option over another based on price: $a_i = 0, b_i = 0$ for $i \neq 5$, $a_5 = -1, b_5 = 300$. The third user model is that of a business traveler that prefers early flights and flights on SAS. As we shall see later this model is able to induce a tradeoff between options (e.g. early non-SAS flights vs. later SAS flights): $a_i = 0, b_i = 0$ for $i < 3, i > 4$, $v_3(\mathsf{'SAS'}) = 1$, $v_3(\mathsf{'Lufthansa'}) = 0.1$, $v_3(\mathsf{'AirFrance'}) = 0.1$, $v_3(\mathsf{'BritishAir'}) = 0.1$, $a_3 = 1$, $b_3 = 0$, $a_4 = -.2$, $b_4 = 2.6$.

We assume the same model of discernibility for all users. For flight number and aircraft type we assume no ability of the users to discern between options, that is $\zeta_1(v_1, v_2) = 1$ and $\zeta_7(v_1, v_2) = 1$ for all value pairs v_1 and v_2. For destination we assume perfect discernibility, that is $\zeta_2(v_1, v_2) = 1$ when $v_1 = v_2$ and 0 otherwise. For airline and meal we assume strong discernibility, specifically $\zeta_3(v_1, v_2) = 1$ and $\zeta_6(v_1, v_2) = 1$ when $v_1 = v_2$ and .33 otherwise. For departure time and price we use an exponential measure. That is $\zeta_4(v_1, v_2) = e^{-|v_1 - v_2|}$ and $\zeta_5(v_1, v_2) = e^{-\frac{|v_1 - v_2|}{100}}$. Given this model, $\mathsf{sim}(t_1, t_4) = .074$.

4.2 System runs

Give the database of table 1 and the user and discernibility models above, table 2 shows the calculation of gain for various clusterings of the input query "the flights to Berlin leaving before noon." The highest benefit clusterings are presented along with several lower scoring alternatives to illustrate the sensitivity to the given user model. The key parameter that controls the behavior of the system is the penalty parameter γ.

5 Discussion

The work presented here is preliminary, likely to undergo much revision and refinement as it is integrated and evaluated within an operational NLI system [2]. Among the unsettled issues are the form and scope of the models of utility and discernibility. For example the independence assumption made in the current model of discernibility is likely to be inadequate in general. As an anonymous reviewer points out, a possible reason why SAS and Lufthansa are discernible could be that SAS provides meals while Lufthansa does not. We agree, although we note that a more sophisticated model could be developed and plugged into our basic approach. As for scope, we have assumed that preference and discernibility models can be crafted for large classes of users in a given context. For example we assume that under a wide variety of conditions flights on A320's and flights on A300's are indiscernible. Likewise we model flights to different locations as nearly perfectly discernible.

While perhaps our models of preference and discernibility should be generalized, we feel justified in stipulating the ideal answer assumption and our method of calculating information gain, confounded by discernibility and preference. A natural question is whether the ideal answer assumption can be relaxed. Our hypothesis is that the discernibility model and γ, the penalty of picking a non-ideal option, already provide enough machinery to get desired results. In any case, some variant of the ideal answer assumption seems necessary to cast the problem in an information theoretic light.

In contrast to *summarize-and-refine* based approaches [4], *user-modeling* based approaches, as characterized by [3, 6] rank matching options based on their utility, offering the highest ranking options first. Recently these two strategies (*summarize-and-refine* and *user-modeling*) have been combined into a single approach [1] based on building an *option tree* over the (current) set of options which specifies refinement paths based on a user model of attribute importance and attribute value preferences. Such option trees are further pruned based on dominance relations amongst options (i.e. when one option will always be preferred over another) and option trees are able to express trade-offs among options when the user has conflicting preferences (e.g. if the user prefers early flights and flights on SAS, then a response might be "At 8 am, flight 2 is the earliest flight to Berlin, but it's with Lufthansa, while flight 7, leaving at 10am, is the earliest flight to Berlin on SAS.") An elusive goal of the work in this paper, not yet achieved, is to provide an information theoretic account of why presenting such trade-offs yields especially high information gain.

6 Conclusions

We live in a time of tremendous choice; we pick from hundreds of mobile phone models, thousands of travel destinations and millions of potential chat partners. When confronted with such complex choices, people tend to become either *maximizers*, spending large amounts of time studying the various options, their features and trade-offs, or

user	response	penalty (γ)	benefit
M. Entropy	"The 2 flights with Lufthansa or 2 flights with SAS?"	3 bits	.39
	"Flight #7 (SAS at 10) or flight #8 (SAS at 11) or flight #2 (Lufthansa at 8)"	3 bits	.25
Student	"Flight #8 the cheapest or the 3 other more expensive flights."	3 bits	.52
	"Flight #8 the cheapest."	.1 bits	1.94
	"Flight #8 the cheapest or flight #5."	.1 bits	.89
Business	"Flight #2 (the earliest), Flight #7 (the earliest on SAS), or the remaining 2 flights?"	3 bits	.51

Table 2: Response to a request for "the flights to Berlin leaving before noon."

they become *satisfiers*, making snap decisions, often bad, but saving time and mental energy. This paper serves both these types through increasing the efficiency of finding high quality options. This paper has presented a method to uniformly measure clusterings that either generalizes the user's query or specializes the user's query or in fact some combination of such strategies. The higher the penalty parameter γ, the more the system will opt toward a maximizer strategy.

This paper follows in the tradition of *cooperative query answering* which seeks to provide the user with more natural answers. This paper has mainly developed a set of theoretical tools and have verified the reasonableness of the developed tool through a simple, distributional implementation of the said concepts. The work in this paper tends more toward *summarize-and-refine* methods than user modeling based techniques. One aspect that the system does not explore are the subtle issues of contrast and linguistic nuance in presenting results. The system follows the *summarize-and-refine* approach in this respect, providing relatively straight forward summarizations of the best clustering that are found. Future work aims toward building a more efficient algorithm to search for possible clusters and incorporating the work into a query paraphrasing and natural language interface system for end-user evaluation [2].

7 Acknowledgements

We thank Johanna Moore and Joseph Palifroni for a fruitful discussion of an earlier version of this work and Jutta Langel and Jurg Kohlas for reminding us of the power and elegance of information theory applied to databases, or, in their case, Information Algebras.

References

[1] V. Demberg and J. Moore. Information presentation in spoken dialogue systems. In *Proc of EACL*, pages 65–72, Trento, Italy, April 2006.

[2] M. Minock. A STEP towards realizing Codd's vision of rendezvous with the casual user. In *33rd International Conference on Very Large Data Bases (VLDB)*, Vienna, Austria, 2007. Demonstration session.

[3] J. Moore, M. Foster, O. Lemon, and M. White. Generating tailored, comparative descriptions in spoken dialogue. In *Proc. of the Seventeenth International Florida Artificial Intelligence Research Sociey Conference*. AAAI press, 2004.

[4] J. Polifroni, G. Chung, and S. Seneff. Towards automatic generation of mixed-initiative dialogue systems from web content. In *Proc. of Eurospeech '03*, pages 193–196, 2003.

[5] M. Walker, R. Passonneau, and J. Boland. Quantitative and qualitative evaluation of darpa communicator spoken dialogue systems. In *Meeting of the Association for Computational Linguistics*, pages 515–522, 2001.

[6] M. Walker, S. Whittaker, A. Stent, P. Maloor, J. Moore M., M. Johnston, and G. Vasireddy. Generation and evaluation of user tailored responses in multimodal dialogue. *Cognitive Science*, 28:811–840, 2004.