Thinking Positively - Explanatory Feedback for Conversational Recommender Systems

Kevin McCarthy, James Reilly, Lorraine McGinty, and Barry Smyth

Adaptive Information Cluster, Smart Media Institute,
Department of Computer Science, University College Dublin (UCD), Ireland.
{firstname.lastname}@ucd.ie

Abstract. When it comes to buying expensive goods people expect to be skillfully steered through the options by well-informed sales assistants that are capable of balancing the user's many and varied requirements. In addition users often need to be educated about the product-space, especially if they are to come to understand what is available and why certain options are being recommended by the sales-assistant. The same issues arise in interactive recommender systems, our online equivalent of a sales assistant and explanation in recommender systems, as a means to educate users and justify recommendations, is now well accepted. In this paper we focus on a novel approach to explanation. Instead of attempting to justify a particular recommendation we focus on how explanations can help users to understand the recommendation opportunities that remain if the current recommendation should not meet their requirements. We describe how this approach to explanation is tightly coupled with the generation of compound critiques, which act as a form of feedback for the users. And we argue that these explanation-rich critiques have the potential to dramatically improve recommender performance and usability.

1 Introduction

In this paper we are interested in how explanations can be used to support the user in an interactive CBR system, in this case a conversational recommender system in which the user is guided through a product space through a number of recommendation cycles [8]. During each cycle a particular product case is recommended to the user based on their initial query and feedback to date; critiquing is used as the basic form of feedback. In this type of recommender system the user is essentially being helped to navigate through a complex product-space—based on the feedback of the user the recommender system chooses a case to suggest during the next cycle in the hope that this case will be closer to that which the user is ultimately looking for. From a user's perspective it is often difficult to know how best to critique a proposed case in order to make useful progress through the product space.

For example, consider a recommender system for Digital Cameras and suppose the user is presented with a *300 euro* model with a *2 mega-pixel* resolution, *64MB* of card-memory and *3x* optical zoom. Let's suppose that the user is actually looking for a greater resolution and more memory, but at a lower price. Ordinarily she would start by critiquing one of the features, *price, resolution or memory.* There may be no cameras to satisfy all three of these constraints and by requesting a camera that is *cheaper* she may find that she is recommended a new camera that is cheaper but one that has lower resolution and less memory, which may not be satisfactory.

The point is that normally this type of conversational recommender system doesn't help the user to understand what product options exist beyond the current case, and this can mean that the user spends time following false-leads and backtracking as she tries different sequences of critiques. We believe that explanations can play a useful role in this regard by providing the user with an indication of the type of products that remain available. For instance, in the above example it might be useful for the system to highlight that there are 30 products with a lower price and greater resolution, but that these products also have less memory. Or that there are 15 products with greater resolution and more memory but they are also more expensive.

In this paper we describe how these type of explanations can be generated, during each recommendation cycle, by using the Apriori data-mining technique to identify and rank frequently occurring sets of features that differ relative to the current case. An added benefit of this approach is that the resulting explanations serve as collections of critiques that can be used by the user as a more effective form of feedback, *Dynamic Critiquing.* Each explanation serves as a *compound critique* covering multiple features. For example, the compound critique $\{[Price <], [Resolution >], [Memory <]\}$ indicates that there are remaining cases available that are cheaper and with a higher resolution but lower memory. By selecting this critique the user can focus their search more effectively and we have previously demonstrated the potential for significant improvements in recommendation efficiency[13]. In this particular paper we focus on the explanatory power of these compound critiques and present examples from a prototype application to illustrate the potential practical benefits of our approach.

2 Background

The use of explanations in case-based reasoning systems is not new, although the role of explanations within CBR systems has evolved over the years. Some of the early work in this area has focused on the use of explanation-based techniques in order to drive the CBR process model. In this sense explanation structures are generated to fulfill various functions with a CBR system: the explanations are constructed by the system, for the system. For example, [2] describe how cases can be be indexed by the use of explanation structures. [1] describe their explanation-driven architecture in which explanation plays a fundamental role in all parts of the CBR cycle. For example, explanations are generated in order to help the system evaluate the goodness of a match between the target problem and any source cases identified as relevant by a superficial, surface-level matching function. During reuse, explanations are used to evaluate the proposed solution and to justify changes to this solution. And during learning, explanations are used to justify whether the newly solved cases should be learned, to determine which part of the case should be learned, and to evaluate the predictive strengths of the case findings. Similarly, the work of [7] describes a case-based planning system that is able to exploit planning failures to improve case retrieval by using explanation-based learning techniques to attempt to explain the planning failure in order to identify causal features that are likely to be predictive of similar failures in the future.

More relevant here is the use of explanations for the benefit of the user by developing systems that are capable of explaining the reasoning steps and conclusions. For example, diagnosis and classification systems may generate explanations in order to justify the outcome of a particular diagnosis or classification in order to satisfy the user of its validity. Alternatively, a product recommender may use explanations to explain the reasons why a particular product was recommended, or indeed why a product was not recommended [11]. In recent years there has been an increasing interest in the use of cases as a source of explanation in these type of situations. Indeed there is considerable optimism among the CBR community regarding the potential value of case-based or *precedent-based* approaches to explanation, when compared to their more traditional rule-based counterparts. The argument has been made that past cases provide a more natural and convincing form of explanation. For example, Cunningham *et al.* [4] report how live-users find similar cases to be more convincing than rules-based explanations in a classification task.

However, using the most similar case in order to justify or explain a particular classification outcome to the user is perhaps the simplest form of case-based explanation and many issues remain a source of active research within the community. Recently, the work of Doyle *et al.* [5] demonstrate that in some classification tasks presenting the nearest-neighbour case may not be the best way to explain or justify a particular classification outcome. This occurs when the nearest-neighbour happens to be farther from the decision surface than the target case, and demonstrate how superior explanation cases can be selected by using an explanation utility metric that formalises this idea.

Presenting the user with an *explanation* case is just the beginning of the explanation process and on its own may not be sufficient to convince the user that a particular diagnosis or classification outcome is justified. For instance, as McSherry points out [9, 10], attempting to justify a predicted outcome by presenting the user with the particular explanation case (whether it is chosen because it was a nearest-neighbour to the target or because of some alternative strategy), ignores the possibility that some of the features of this explanation case may conflict with some of the target features to oppose rather than support the predicted outcome. This may mislead the user if they mistakenly view these opposing features to be evidence in favour of the predicted outcome [9]. To combat this problem, McSherry proposes an evidential approach to precedent-based explanation [9, 10]. The user is presented with evidence that both supports and opposes a particular outcome in order to explain the pros and cons of case-based conclusions.

Of course sometimes a system may not be able to produce a satisfactory outcome, so instead of justifying a conclusion, the system must attempt to explain the cause of the reasoning failure. A good example of this is presented in [11], which describes a mixed-initiative approach to recovery from retrieval failure by helping the user to eliminate certain constraints from their initial queries in a conversational product recommender. An important aspect of this work is that explanations are generated in order to explain retrieval failures by highlighting subsets of query features that cannot be satisfied (eg. *"there are no cameras with price less than 300 euro and resolution greater than 4 mega-pixels"*. This is particularly relevant and complimentary to the work described in this paper if one considers a recommendation cycle where the user has not found their target product to be a type of retrieval failure. In our approach, instead of trying to explain the retrieval failure we attempt to explain the retrieval opportunities that remain (eg. *"there are 10 cameras less than 300 euro but their resolution is between 1 and 4 mega-pixels"* or *"there are 20 cameras with between 4 and 6 mega-pixels but their price is more than 300 euro"*). To put this another way: instead of explaining what types of products do not exist, we explain what types of products do exist. We argue that this type of *positive* explanation is likely to be more useful as it is more intuitive for users to respond to explanations that tell them what they can do rather than what they can't do.

3 Compound Critiques as Explanations

In this work we assume a conversational recommender system in the image of the well-known *FindMe* recommender systems [3]. Each recommendation session is initiated by an initial user query and this results in the retrieval of the most similar case (*the recommended case*) and a set of critiques, both fixed and dynamic. The user will have the opportunity to accept this case, thereby ending the recommendation session, or to critique it. If they critique the case, the critique acts as a filter over the remaining cases, such that the case chosen for the next cycle is the one that is compatible with the critique and maximally similar to the previously recommended case.

Our contribution is to generate a set of compound critiques during each recommendation cycle and to present promising compound critiques to the user for consideration. Each compound critique covers a number of features (typically 2 or 3) and they are generated in realtime based on the cases that remain for a particular cycle.

3.1 Critique Patterns

In order to discover a useful set of compound critiques from a set of product cases we need to first understand how these cases relate to the current recommendation—we need to understand their basic feature differences—so that we may then look for frequently occuring sets of differences to act as our compound critiques. The first step is to generate a set of so-called *critique patterns* from the remaining cases. Each remaining case is compared to the current recommended case and the relative feature differences make up the critique pattern. For example, the critique pattern in Figure 1, which is taken from the digital

	Current Case	**Case c from CB**	**Critique Pattern**
Manufacturer	Kodak	Nikon	!=
Optical Zoom (x)	4	3	<
Memory (MB)	256	64	<
Weight (Grams)	292	348	>
Resolution (M Pixel)	4.6	3.2	<
Size	Medium	Large	!=
Case	Magnesium	Aluminium	!=
Price	450	385	<

Fig. 1. Critique patterns explicate the differences between the current recommendation and the remaining cases.

camera domain, reflects how a candidate case c differs from current case in terms of individual feature critiques. For instance, the critique pattern shown includes a "$<$" critique for *optical zoom*— we will refer to this as $[Zoom <]$—because the comparison case has a lower optical zoom than the current recommended case. Similarly, the pattern includes the critique $[Weight >]$ because the comparison case is heavier than the current case.

3.2 Mining Compound Critiques

To identify compound critiques we must look for recurring patterns within a set of critique patterns and this can be efficiently achieved using the well-known Apriori algorithm[6, 12]. Apriori characterises recurring item subsets as association rules of the form $A \rightarrow B$: from the presence of a certain set of critiques (A) one can infer the presence of certain other critiques (B). For example, one might learn that from the presence of the critique, $[Price <]$, we can infer the presence of $[Resolution <]$ with a high degree of probability; in other words the pattern $\{[Price <],[Resolution <]\}$ is commonplace.

Apriori measures the importance of a rule in terms of its *support* and *confidence*. Support is the percentage of patterns for which the rule is correct; that is, the number of patterns that contain both A and B divided by the total number of patterns. Confidence, on the other hand, is a measure of the number of patterns in which the rule is correct relative to the number of patterns in which the rule is applicable; that is, the number of patterns that contain both A and B divided by the number of patterns containing A.

Our proposal is to use Apriori, during each recommendation cycle, to generate a collection of compound critiques (frequent itemsets over the pattern-base), and to then select a small subset of the *best* of these compound critiques for presentation to the user to complement the standard unit critiques. In *Dynamic Critiquing*[13] we evaluate a number of alternative selection strategies to find that selecting compound critiques with low support values delivers superior results—critiques with low support values have the potential to eliminate many cases from consideration.

4 Example Session

Figures 2-4 presents a series of screenshots from a prototype application of our dynamic critiquing approach in the context of an online digital camera store. The screenshots present a sequence of recommendation cycles and in each we see the currently recommended case, its features and their unit critiques, plus a set of 3 compound critiques, and their associated explanations. Each compound critique is translated into an English explanation and can be chosen directly (via the 'pick' option) or elaborated further (via the 'explain').

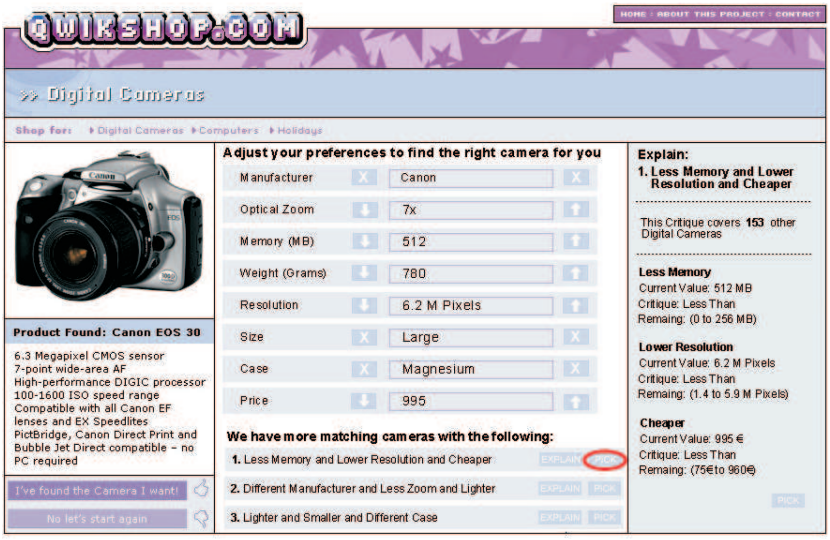

Fig. 2. Initially the user is presented with a high-end Canon camera but selects the first compound critique to indicate that they are looking for something cheaper and are willing to come down on memory and resolution.

After the user has provided some initial information they are presented with a high-end Canon camera for 995 euro with 512MB of memory and 6.2 mega-pixels, as shown in Figure 2. The user can critique any of the individual features, such as *manufacturer, memory, resolution* etc. by selecting the appropriate critique icon on either side of the feature value fields that are displayed for the current camera; the up-arrow indicates a greater-than critique, down-arrow indicates a less-that critique and the cross indicates a not-equal-to critique. In addition, just below these features, three compound critiques are displayed and the user can either select one of these to be applied directly (the 'pick' option) or they can request a more detailed explanation via the 'explain' option. This explanation is presented in the pane to the right of the feature values, and defaults to be an explanation of the first compound critique.

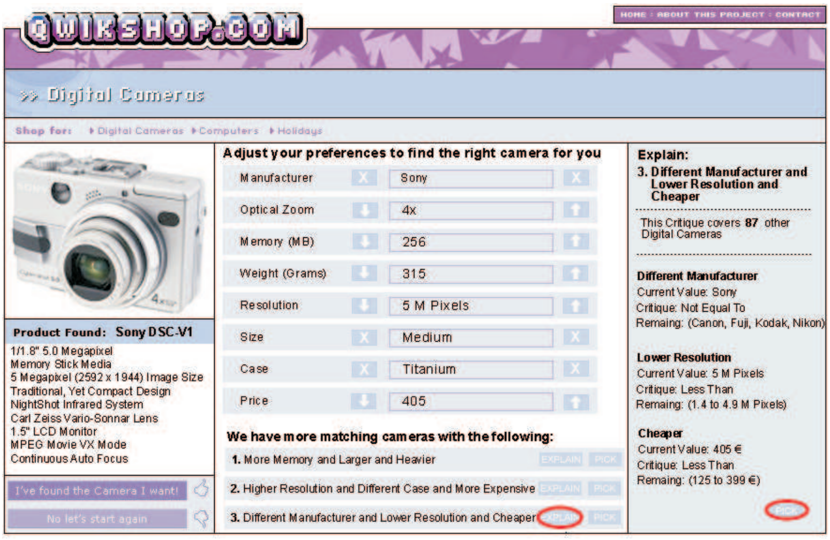

Fig. 3. This camera is cheaper with less memory and a lower resolution. It is still too expensive though, but the compound critique tells the user that there are cheaper cameras by different manufacturers if she is willing to go with a lower resolution. The corresponding explanation convinces the user that there are reputable manufacturers remaining and that the lower resolutions offered are still acceptable given the significant price drop that is available. The user chooses to go with this compound critique.

For example, in Figure 3 the user asks for further explanation of the third compound critique (*"Different Manufacturer, Lower Resolution and Cheaper"*). The resulting explanation tells the user that there are 87 remaining cameras that satisfy this critique—that is, there are 87 cameras that are cheaper, with a lower resolution, and made by a different manufacturer, than the currently recommended Sony camera. In addition, the explanation provides information about the ranges of values for these critiqued features. For instance, the user is told that these 87 cameras are made by manufacturers such as Canon, Fuji, Kodak and Nikon, that they have resolutions from 1.4 to 4.8 mega-pixels, and that their price ranges from 125 to 399 euro.

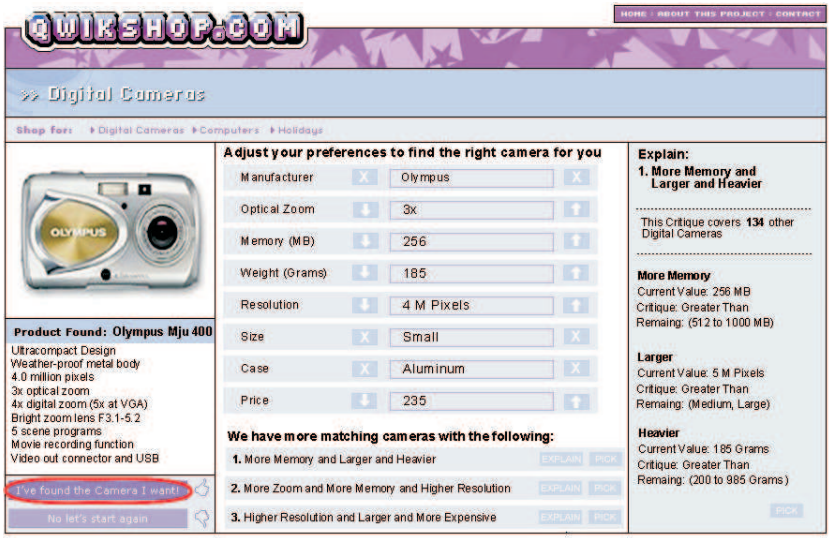

Fig. 4. Finally, the user is recommended a camera that they are happy with: a 235 euro Olympus with a respectable resolution and an acceptable memory capacity.

5 Discussion

Our original motivation for embarking on this strand of research was brought about by the desire to improve the efficiency of critiquing as a form of recommendation feedback. We feel that critiquing is particularly well adapted to many consumer oriented e-commerce applications but it can lead to protracted recommendation sessions, especially if users end up following many false leads, which is always a risk in the absence of guidance from the system. In related research [13] we have already demonstrated the potential of our dynamic critiquing approach when it comes to improve recommendation efficiency.

During the development of a prototype application to demonstrate dynamic critiquing in a realistic application context, it became clear that the compound critiques also had a potentially valuable role to play in explanation, which is the main point of this paper. The core hypothesis is that compound critiques help the user to better understand the *recommendation opportunities* that exist beyond the current cycle by helping them to appreciate common interactions between features. We believe that in many recommender domains, where the user is likely to have incomplete knowledge about the finer details of the feature-space, that compound critiques will help to effectively map out this space. For this reason we believe that users will actually find it easier to work with compound critiques, and their associated explanations, than unit critiques and this may, for example, help the user to make fewer critiquing errors. For instance, with standard critiquing in the digital camera domain a user might naively select the $[Price <]$ unit critique in the mistaken belief that this may deliver a cheaper camera that satisfies all of their other requirements. However, reducing price in this way may lead to a reduction in resolution that the user might not find acceptable and, as a result, they will have to backtrack. This problem is less likely to occur if the compound critique $\{[Price <], [Resolution <]\}$ is presented because the user will come to understand the implications of a price-drop prior to selecting any critique.

At this point we are at the stage where the core technology has been developed and is being deployed in a realistic application setting. In addition the evaluation described in [13] provides evidence from an off-line study that if compound critiques are selected by users then their potential to improve recommender performance is significant. However, this approach has not been tested on live-users and thus we cannot say whether or not these benefits will be available in practice. For example, if the compound critiques presented are not relevant to users then they are unlikely to be chosen. For this reason we need to fully evaluate how relevant these critiques are in live-user trials. Similarly, even if the compound critiques are relevant it is not clear whether users will come to fully appreciate their implications, even using the elaboration features described above. Once again, live-user trials are required.

Finally it is worth pointing out, of course, that our approach to explanation is but one of many different approaches to explanation. We have focused on the need to help users to understand what options remain available, if the current recommendation should not meet their requirements. As we have seen in Section 2, other approaches to explanation have a different focus, such as justifying a particular recommendation or explaining its pros and cons. In the future, it is likely that we will come to see many of these explanation strategies playing their own particular roles in the next generation of interactive recommender systems.

6 Conclusions

Explanations have an important role to play in helping users to understand the suggestions made by recommender systems. Much of the research conducted to date has focused on ways to justify a particular recommendation to the end-user. In this paper we have taken a different stance by proposing a technique that is designed to help the user to understand the recommendation opportunities that exist beyond the current suggestion, on the assumption that this current suggestion does not satisfy all of the user's implicit requirements. We have described a technique for automatically generating compound critiques and explained how these compound critiques can be used as rich explanations as well as providing the user with direct feedback opportunities.

Our work has been implemented in a live demonstration system and off-line evaluations have demonstrated the potential for significant performance improvements over standard recommendation techniques. Going forward we aim to conduct a series of live-user trials in order to better gauge the response of live-users to different types of compound critiques and explanations.

References

1. A. Aamodt. Explanation-Driven Case-Based Reasoning. In S. Wess, K.D. Althoff, and M. Richter, editors, *Proceedings of the European Workshop on Case-Based Reasoning (EWCBR-94)*. Springer, 1994.
2. R. Barletta and W. Mark. Explanation-Based Indexing of Cases. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88)*. AAAI, 1988. Minneapolis, MN, US.
3. R. Burke, K. Hammond, and B.C. Young. The FindMe Approach to Assisted Browsing. *Journal of IEEE Expert*, 12(4):32–40, 1997.
4. P. Cunningham, D. Doyle, and J. Loughrey. An Evaluation of the Usefulness of Case-Based Explanation. . In K. Ashley and D. Bridge, editors, *Case-Based Reasoning Research and Development. LNAI, Vol. 2689.*, pages 191–199. Springer-Verlag, 2003. Berlin.
5. D. Doyle, P. Cunningham, D. Bridge, and Y. Rahman. Explanation Oriented Retrieval. In P. A. Gonzalez Calero and P. Funk, editors, *Proceedings of the European Conference on Case-Based Reasoning (ECCBR-04)*. Springer, 2004. Madrid, Spain.
6. Z. Hu, W.N. Chin, and M. Takeichi. Calculating a New Data Mining Algorithm for Market Basket Analysis. *Lecture Notes in Computer Science*, 1753:169–175, 2000.
7. L. Ihrig and S. Kambhampati. An Explanation-Based Approach to Improve Retrieval in Case-Based Planning. In M. Ghallab and A. Milani, editors, *Proceedings of the European Workshop on Planning (EWSP-95)*, pages 395–406. IOS Press, 1995. Amsterdam.
8. L. McGinty and B. Smyth. Tweaking Critiquing. In *Proceedings of the Workshop on Personalization and Web Techniques at the International Joint Conference on Artificial Intelligence (IJCAI-03)*. Morgan-Kaufmann, 2003. Acapulco, Mexico.
9. D. McSherry. Explanation in Case-Based Reasoning: An Evidential Approach. . In B. Lees, editor, *Proceedings of the 8th UK Workshop on Case-Based Reasoning*, 2003. Cambridge, UK.
10. D. McSherry. Explaining the Pros and Cons of Conclusions in CBR. . In P. A. Gonzalez Calero and P. Funk, editors, *Proceedings of the European Conference on Case-Based Reasoning (ECCBR-04)*. Springer, 2004. Madrid, Spain.
11. D. McSherry. Incremental Relaxation of Unsuccessful Queries. . In P. A. Gonzalez Calero and P. Funk, editors, *Proceedings of the European Conference on Case-Based Reasoning (ECCBR-04)*. Springer, 2004. Madrid, Spain.
12. R. Srikant H. Toivonen R. Agrawal, H. Mannila and A. Inkeri Verkamo. Fast Discovery of Association Rules in Large Databases. *Advances in Knowledge Discovery and Data Mining*, pages 307–328, 1996.
13. J. Reilly, K. McCarthy, L. McGinty, and B. Smyth. Dynamic Critiquing. In P. A. Gonzalez Calero and P. Funk, editors, *Proceedings of the European Conference on Case-Based Reasoning (ECCBR-04)*. Springer, 2004. Madrid, Spain.