COPE: Interactive Image Retrieval Using Conversational Recommendation

Bartoz Balcer
University of Glasgow
Glasgow G12 8QQ
0704842B@student.gla.ac.uk

Martin Halvey
Glasgow Caledonian University
Glasgow G4 0BA
martin.halvey@gcu.ac.uk

Stephen A. Brewster
University of Glasgow
Glasgow G12 8QQ
stephen.brewster@glasgow.ac.uk

Joemon M. Jose
University of Glasgow
Glasgow G12 8QQ
joemon.jose@glasgow.ac.uk

Most multimedia retrieval services e.g. YouTube, Flickr, Google etc. rely on users searching using textual queries or examples. However, this solution is inadequate when there is no text, very little text, the text is in a foreign language or the user cannot form textual a query. In order to overcome these shortcomings we have developed an image retrieval system called COPE (CONversational Picture Exploration) that can use a number of different preference feedback mechanisms, inspired by conversational recommendation paradigms, for image retrieval. In COPE users are presented with a small number of search results and simply have to express whether these results match their information need. We examine the suitability of a number of feedback approaches for semi-automatic and interactive image retrieval. For interactive retrieval we compared our preference based approaches to text based search (where we consider text to be an upper bound), our results indicate that users prefer preference based search to text based search and in some cases our approaches can outperform text based search.

Keywords: Preference, recommender system, image, retrieval, search non-text

1. INTRODUCTION

Most multimedia retrieval services e.g. YouTube, Flickr etc. rely on users searching using textual queries. However, this places a burden on the users of such systems to be able to describe in a few words the object that they wish to retrieve. Annotations and tags also do not present a complete solution to problems associated with retrieving images. It is usual for users to have different perceptions about the same image and as such will annotate those images differently. This can result in synonyms, polysemy and homonymy, which make it difficult for other users to retrieve the same media objects (Guy and Tonkin 2006). Indeed how retrievable an object is can depends on the quantity of annotations that an object has (Halvey and Keane 2007). Thus using text for retrieval does not work well when the user has a vague idea of what they would like to retrieve or if the multimedia object is not adequately described. In the worst case when an image collection has no annotations or when the annotations are in a foreign language, it is practically impossible for a

user to search the collection. Even in the best case users have difficulty in forming queries due to the semantic gap, i.e. the difference between low level features that machines use to represent multimedia and the high level concepts that humans associate with the same image (Jaimes et al. 2005).

In order to overcome the problems of a lack of text, content-based image retrieval systems can be used, these systems index image data using the visual features of the images e.g. colour, shape, etc. These systems can be difficult to use, as a query must be expressed visually. For example, sketch interfaces (Flickner et al. 1997) force the user to draw the target images required, which can be difficult for some users. Alternatively the user should possess an exemplar image which they submit as a query (query by example).

In an effort to overcome some of the shortcomings highlighted above Villa et al. (Villa et al. 2010) consider an alternative approach, in their approach, a user can search using text in an intermediate database as a way of finding visual examples of their information need. These visual examples can then be used to search a database that lacks

annotations. However this approach assumes the existence of a collection that can be queried via text, which also contains images that have enough similarity to retrieve suitable images. In addition users must be able to differentiate between good query images and relevant images, which might be difficult, especially for novice users.

We present an alternative and also complimentary approach to that of Villa et al. We have developed an image retrieval system called COPE (CONversational Picture Exploration) that can use a number of different preference feedback mechanisms, inspired by conversational recommendation, for preference based image retrieval. Conversational recommender systems attempt to engage the user in an extended recommendation dialog during which the system attempts to produce query information to refine recommendations. In COPE users are presented with a small number of search results and simply have to express whether these results match their information need or not. We examined the suitability of a number of feedback approaches in two experiments, the first for semi-automatic retrieval and the second for interactive image retrieval. In particular for interactive retrieval we compare our preference based approaches with text based search (where text is considered an upper bound).

2. RELATED WORK

2.1 Interactive Image Retrieval

There are a number of innovative image retrieval systems that have attempted to provide interactive tools to overcome the problems associated with image retrieval, which were outlined in the previous section. EGO (Urban and Jose 2006) is a tool for the management of image collections which comprises of a workspace and a recommendation system, these facilities allow for different types of requirements, enabling the user to both search and organise results effectively. With the Ostensive Browser Campbell et al. (Campbell 2000) present a novel image search and browsing system. The main component of the interface is a workspace with objects on it and links between those objects. Candidate images for browsing are determined by an ostensive model, which encompasses a temporal profile of uncertainty. CueFlik (Fogarty et al. 2008) is a Web image search application that allows users to create their own rules for ranking images based on their visual characteristics. Users can then re-rank possible search results according to these rules. The MediaGLOW system (Girgensohn et al. 2009) presents an interactive workspace that allows users to organise photographs. Users can group photographs into stacks in the workspace; these stacks are then

used to create neighbourhoods of similar photographs automatically.

A number of video retrieval systems have also been developed to aid user interaction, although image and video retrieval present slightly different problems, there are also significant overlaps between problems in both areas. The ForkBrowser (de Rooik et al. 2008) embeds multiple search methods into a single interface for browsing. The multiple search methods are presented to the user in the form of threads, visualised in the shape of a fork. These threads are ranked lists of shots based on one of the search methods implemented in the interface. The shot at the top of the stem of the fork is the video that the user is currently viewing, with the tines representing the different threads. The ExtremeBrowser (Hauptmann et al. 2006) aims to maximise the human capability for judging visual material quickly, while at the same time applying active learning techniques using the user selected videos. Videos are presented to the user via rapid serial visual presentation which allows the user to make fast judgements about high numbers of videos. The feedback from the user is used in an active learning loop, which is used to rank the remaining results that the user will review. The FacetBrowser (Villa et al. 2008) is a video search interface that supports the creation of multiple search "facets", to aid users carrying out complex video search tasks involving multiple concepts. Each facet represents a different aspect of the video search task. These facets can be organised into stories by users, facilitating the creation of sequences of related searches and material which together can be used to satisfy a work task. VIGOR (Halvey et al. 2009) is a video retrieval system that allows users to group videos in order to facilitate video retrieval tasks. This grouping allows users to conceptualise their retrieval task and also provides facilities for providing recommendations to users based on these groups.

While our approach shares some characteristics with the systems outlined here, there are some distinct differences. The interaction techniques used in COPE could be considered a form of relevance feedback (Ruthven and Lalmas 2003) i.e. they take the results that are initially returned from a given query and to use information about relevance to perform a new query, and relevance feedback has been used in many other image retrieval systems (Campbell 2000; Urban and Jose 2006). However, in this work we are adapting previous approaches from the recommender system domain for image retrieval in order to exploit the overlap with issues in image retrieval as opposed to web or document retrieval. In addition we are not using any text for our approach; many other approaches use a combination of text and visual features. Finally part of our focus is on the different interactions allowed by conversational recommendation and not simply on the

implementation of the system. Our approach is distinct from many query-by-example systems, in that through interaction the users are reducing the set of possible matching images, as such the system is adapting to user needs and preferences as opposed to providing a new demonstration of the desired characteristic with every query.

2.2 Preference Based Recommendation

Conversational recommender systems attempt to engage the user in an extended dialog during which the system attempts to produce additional information to refine recommendations. The aim is to assist the user in navigating through a complex information space by removing items from consideration based on user feedback. There are two forms of conversational recommender systems, navigation *by asking* or navigation *by proposing* (Shimazu et al. 2002). In the case of the navigation by asking, the system asks the user a series of questions regarding their requirements; this form of feedback is sometimes referred to as *value elicitation*. On the other hand, systems that employ *navigation by proposing* avoid asking the user direct questions, but instead present users with interim recommendations and ask for feedback, usually in the form of a simple preference or a rating. COPE utilises navigation by proposing.

2.2.1 Navigation by Proposing

In navigation by proposing the user is presented with one of more alternatives during each interaction cycle. The user is then asked to offer feedback in relation to these alternatives and the interaction cycles continue. There are 3 main types of feedback that can be used. The simplest is *preference based feedback*, during each cycle the user simply expresses a preference for one alternative. Another approach is expressed in the form of a constraint over certain features of one of the recommendations; this is commonly known as *critique-based feedback*. Finally *ratings-based feedback* involves the user providing an explicit rating of a recommendation. In this work we are not using rating based feedback as it places a burden on users to provide subjective feedback, rather than the more simple feedback afforded by the other approaches.

2.2.1.1 Preference Based Feedback

Preference based feedback is extremely suitable for tasks where users have very little domain knowledge, but can easily express a preference. Thus this approach to conversational navigation should be well suited for image retrieval. However, while this approach carries very little feedback overhead for the user, it can be limited in its capacity to guide the interaction process, i.e. it may not always be clear why a user has selected one object over another, both may have many common

features and many distinguishing features. In an attempt to address this issue the *comparison-based recommendation* work of McGinty and Smyth (McGinty and Smyth 2002) propose a number of query revision strategies that are designed to revise the current query as a result of preference based feedback. The most straightforward of these strategies (*more like this*) simply adopts the preferred case as a new query and proceeds to retrieve the k most similar cases to it for the next cycle. However, this approach might not be very efficient as it does not really attempt to infer the user's true preferences at a feature level. A second approach (*partial more like this*) transfers features from the preferred case, if these features are absent from all of the rejected cases, thus allowing the system to focus on those aspects of the preferred cases that are unique in the current cycle. A third strategy attempts to weight features in the updated query according to how confident the recommender can be that these features are responsible for the user's preference (*weighted more like this*). One particular weighting strategy depends on the number of alternatives for a given feature within the current recommendation set. A final strategy was to allow users to give return a case as negative feedback to retrieve cases less like the returned case (*less like this*).

2.2.1.2 Critique Based Feedback

Critiquing-based recommenders allow users to provide feedback in the form of a directional feature constraint e.g. asking for a TV that is *bigger* than the current TV, *bigger* being a critique over the *size* feature. The feedback mechanism is one that does not require the user to have extensive domain knowledge. The Entree recommender (Burke et al. 1996) suggests restaurants in Chicago and each recommendation allows the user to select from seven different critiques. When a user selects a critique such as *cheaper*, Entree eliminates cases (restaurants) that do not satisfy the critique from consideration in the next cycle, and selects that case which is most similar to the current recommendation from those remaining; thus each critique acts as a filter over the cases. In the Car Navigator recommender (Shimazu 2001) system, individual critiques were also designed to cover multiple features, so that, for instance, a user might request a *sportier* car than the current recommendation, simultaneously constraining features such as *engine size* and *acceleration*. These *compound critiques* obviously allow the recommender to take larger steps through the information-space, eliminating many more cases than would be possible with a single-feature, *unit critique*, in a single recommendation cycle. Recently the work of McCarthy et al. (McCarthy et al. 2004; McCarthy et al. 2005) has investigated the possibility of automatically generating dynamic compound critiques based on the remaining cases

and the user's progress so far. This *dynamic critiquing* approach uses data mining techniques to identify groups of unit critiques that reflect common difference patterns between the remaining cases.

3. IMAGE COLLECTION

For our experiments the CLEF 2007 image collection was chosen (Grubinger 2007; Grubinger and Clough 2007). CLEF 2007 is a set of 20,000 images, 60 search topics, and associated relevance judgments. As part of the CLEF 2006 effort, which shared the same set of topics as used in CLEF 2007, the topics were categorised into a number of different categories, including: easy/hard, semantic/visual, and geographic/general. In this current work, five different visual features are used to represent each image:

- Colour Layout: the spatial distribution of colours in an image
- Colour Histogram: colour distribution of the image
- Edge Histogram: the spatial distribution of the edges in the image
- Homogeneous Texture: region texture representation
- Colour Structure: represents an image by a combination of colour distribution and the local spatial structure of the colour

The extraction and representation of each of these features is described in (Manjunath et al. 2002).

4. SIMULATED STUDY

4.1 Implemented Approaches

4.1.1 Compounding Critiquing

The main problem with applying Compound Critiquing to image retrieval was finding a convenient and efficient way of creating unit critiques that would start off the algorithm. In other domains the methods for finding and defining feature patterns are largely self-defining and rather straight-forward. For example, consider a system where the user can browse digital cameras for sale (Shimazu et al. 2002). All products have atomic features like price, manufacturer, resolution etc. Unit critiques can be easily constructed in this example using Boolean values or inequality operators e.g. 'higher price', 'different manufacturer' etc. For images it is much harder to define features of equally low complexity which could still hold some meaningful information. Instead, it is normal for some kind of multidimensional vectors to be used to describe visual information e.g. colour histograms, edge

histograms etc. In order to overcome this problem the following solution was adopted:

- Separately for each feature, distance to the current prototype is computed and a sorted list of images is created using these results.
- For each feature the corresponding list is split into two sub lists, where the dividing point is determined by multiplying the amplitude, i.e. the biggest distance minus the smallest distance, by a fraction which is called the *split factor*.
- The sub lists for each feature become two unit critiques. One sub list can be described as more similar to the prototype or as being smaller than the split point. The other sub list is considered as less similar or as being bigger than the split point.

Figure 1: Operation of compound critiquing approach

If we assume that the split factor is set to 1/2, then one sub list will contain only elements with distances smaller than a half of the amplitude and the other will contain only elements with distances greater than a half of the amplitude. A more naive way would be to just split the initial list into two equal halves but this would give more random combination of items in the sub lists. Also this would disregard the distribution of distances completely, whereas the split factor at least guarantees the items in each sub list will hold some properties that can be expressed mathematically. Conversely, more sophisticated methods based on classification or clustering algorithms could be used, but given that the application is supposed to engage the user in an extensive dynamic dialogue with many cycles it is preferable for now to use

simpler and faster techniques. Future work could look at more sophisticated approaches for creating critiques; however, this is beyond the scope of the work presented here.

The second part of the Compound Critiquing algorithm involves creating association rules from the unit critiques which are later turned into compound critiques. In general, association rules are used to mine 'interesting combinations of features' or popular patterns and they focus on examining the frequency of similar values for some features appearing together. We choose Apriori (Agrawal and Srikant 1994) to generate association rules. It is one of the most popular and relatively simple techniques which is based on calculating the support (frequency) for each rule and creating complex rules from lower-order ones. Association rules were created as follows:

- Previously created unit critiques with support below the defined threshold are discarded.
- Remaining unit critiques are combined with each other to create 2-feature critiques. The new critiques have their elements sorted by the sums of the distances for both features.
- New critiques that are below the support threshold are discarded.
- 3-feature critiques are created by combining 2-feature critiques with unit critiques and the process is iteratively repeated to create higher-order critiques as long as new ones can still be constructed.

Eventually, the algorithm finishes by returning a list of critiques which all have support levels exceeding the threshold. Apriori in the worst case can have an exponential complexity. For this work this is considered acceptable, since there are only 5 features defined for all images and asymmetric splitting of initial lists of distances, using the split factor, makes it more likely that some critiques will have insufficient support and will be rejected.

4.1.2 xMLT Approaches

Three of the approaches proposed by McGinty and Smyth (McGinty and Smyth 2002), namely More-Like-This (MLT), partial More-Like-This (pMLT) and weighted More-Like-This (wMLT), were implemented as proposed query processing methods. These approaches were described as "promising" variants in the original paper. The other two, LLT (Less-Like-This) and MLT+LLT (combination of MLT and LLT) were removed from the consideration currently, as expressing negative feedback is an additional variable that we reserve for investigating in future work. Also as outlined by McGinty and Smyth LLT is also very difficult to combine with other approaches (McGinty and Smyth 2002).

The operation of the MLT algorithm is as follows:

- For each image in the available set, normalised distances to the given prototype are computed with respect to each feature. These are then summed to give the overall distance across all features.
- A list is created containing all images sorted by their distances to the prototype.
- A *reduction factor* can optionally be applied to the list of images. This *reduction factor* removes a certain percentage of most dissimilar images, thus preserving some information from the current feedback in future cycles.

The pMLT algorithm has some additional steps compared to MLT:

- Average distances, with respect to each feature, are computed between the chosen prototype and other images that were displayed in the previous cycle.
- A subset of all features which gave the largest average distances is chosen (three features were chosen in the simulations outlined in this Section).
- The remaining operations are essentially the same as the MLT algorithm, but using a smaller subset instead of all features.

Thus, the pMLT algorithm tries to imitate the process of discovering the subset of features that could have attracted the user in a chosen prototype. Similarly to other problems to do with the application of compound critiquing to image retrieval, this is a much more subtle and less obvious task and the solution above is highly dependent on proper computation and normalisation of distances. The differences between the wMLT and pMLT approaches are outlined below:

- Instead of choosing a subset of features, a map of weights is created for the features where each weight is simply the average distance, with respect to a given feature, between the prototype and images ignored by the user.
- When the overall distances are computed for all images, they are multiplied by the corresponding weights and later divided by the sum of all weights to normalise them again.

4.2 Simulated Evaluation Methodology

It was necessary to conduct a simulated study for two reasons. Firstly, we wanted to evaluate if it was possible and useful to apply these approaches to image retrieval. Secondly, many of the implemented algorithms included several parameters which are crucial for their operation. It

was hoped that through a simulation that some of these parameters could be tentatively optimised or tuned. The ImageCLEF collection (see Section 3) includes 60 predefined search topics and for each of the topics three different example images were provided. Moreover, relevance pools were also available listing all images that were relevant for particular topics. Each of the 180 example images was used in turn as an initial query to perform a semi-automated retrieval. Additionally, as each simulation involved random choices (simulating different choices of a human user) three full test runs with the same configuration were conducted. Another reason for choosing the ImageCLEFphoto 2007 collection was the classification of topics (Grubinger 2007; Grubinger and Clough 2007). In this work we used the visual/semantic and difficult/hard classifications.

4.3 Results

4.3.1 Comparison of xMLT Approaches

The three variations of the MLT approach were compared with the reduction parameter set to 50%, i.e. the second half of the ordered result set would be deleted after each cycle. The next prototype was always chosen randomly from the top 5 results returned in the previous cycle. All approaches seemed to work very well for purely visual topics. Overall, the plain MLT approach performed slightly better than the pMLT and wMLT versions. A Wilcoxon signed-rank test showed that the performance difference between the MLT and the wMLT or pMLT approaches in terms of finding relevant results was not significant.

4.3.2 Result Set Reduction in xMLT

The xMLT approaches by default do not automatically reduce the result set. This means that in each cycle the algorithm has to work with all images available in the database, this could be considered to be an example of an infinite browsing strategy. A second set of evaluation runs was conducted to investigate whether the lack of reduction can improve the overall performance of the xMLT algorithms. The results indicated that not reducing the result set gives a small improvement in the case of all MLT approaches and at the same time this did not greatly affect the average number of cycles needed to retrieve desired images. However, the differences between MLT approaches using no reduction and a set reduction factor of 50% were shown to be of no statistical significance and cannot be considered binding.

4.3.3 Support Thresholds in Compound Critiquing

Having critiques that reduce the number of elements in the result set too quickly is not desirable. Test runs were performed with three different values of required support: 40%, 50% and 70%. One critique was chosen randomly in each

cycle without any other restrictions and the split factor used was 1/2 to avoid any bias effect. It was found that the results using lower support values give better results (see Table 1). Setting the threshold to 40% will guarantee finding relevant images for twice as many topics as with a threshold of 70%. Wilcoxon signed-rank tests for successful runs across the different support thresholds indicated that a 40% value is significantly better than 50% ($z=3.221$, $p=0.001$) or 70% ($z=7.071$, $p<0.001$) in terms of finding relevant results.

Table 1: Comparison of support thresholds

Support	Total found in x runs				Cycles
	Any	1	2	3	
40%	122(67.78%)	58	35	29	3.97
50%	99 (55%)	46	25	28	4.02
70%	62 (34.44%)	26	7	29	3.22

4.3.4 Item Split Factors in Compound Critiquing

The split factor provides a way of creating critiques by dividing the set of results into two subsets. Since the subsets are usually of unequal sizes, testing different split factors only makes sense for low support thresholds (with higher thresholds the smaller subset would always be eliminated). A split factor closer to 1 also gives more diversity in the subset containing most similar elements. Five different values for the split factor (1/3, 1/2, 3/5, 2/3 and 3/4) were tested with the support threshold set to 40%. The best results were obtained with split factors in the range between 1/2 and 2/3. On the basis of a Wilcoxon signed-rank test it was discovered that a value of 1/2 is significantly better than a very low split factor of 1/3 ($z=7.288$, $p<0.001$) and that the differences between higher values were not statistically significant for finding relevant results.

4.3.5 Unit vs. Compound Critiques

Critiques may either focus on one feature (unit critiques) or more features (compound critiques). Unit critiques have the advantage of providing more precise feedback to the algorithm, whereas compound critiques should allow for faster convergence. The two types of approach were compared in a series of tests. The results indicate that compound critiques work better than unit critiques for all types of topics. A comparison of the number of successful searches using a Wilcoxon signed-rank test showed that compound critiques perform significantly better than unit critiques across all topic types ($z=3.051$, $p=0.002$).

4.3.6 Features in Compound Critiques

Evaluations were also conducted to discover whether any of the five feature vectors available for all images in the collection demonstrated different properties. In each test, only a critique including the specified feature could be chosen in each cycle (with no other restrictions, i.e. both unit and

compound critiques were allowed). The results indicated that critiques based on colour structure, edge histograms and homogeneous textures seemed to produce the best results. Colour layout was almost equally good. Critiquing based on colour histogram is distinctly worse and fails to find relevant images even for most visual topics. However, the differences in performance for the four best features were shown to be insignificant.

4.4 Design Recommendations

The results of the simulated study had important consequences for the development of COPE. The most important design decision was eliminating the pMLT and wMLT approaches from further consideration, for now, as they essentially worked just like the basic MLT version but were slightly less effective. The reduction factor in MLT was also set at 30% on the basis that reducing the set of results by too much proved to negatively affect the overall performance. The compound critiquing approach will be assigned specific values for the support threshold and split factor. The former was set at 40%, whereas the latter was chosen to be within the range of 1/2 to 3/5 with the exact number to be determined later by trial and error. Moreover, the results allowed us to simplify the critiquing mechanism for human users by eliminating two of the five features available. Colour histograms were discarded as well as colour structures. Colour layouts were chosen in favour of colour structures as they offer comparable performance but are expressed using vectors with a smaller number of dimensions which can speed up calculations. Having only three features instead of five made it possible to create labels for critiques that humans could better comprehend - colour layout, edge histogram and homogeneous texture became "colour", "shape" and "texture", respectively, making them easier for individuals to understand.

5. COPE SYSTEM INTERFACE

Three derivatives of the COPE system were implemented using recommendations from the simulated study (a fourth version was implemented to act as a baseline). Figure 2 shows an example of the user interface that was used for the combination compound critiquing and MLT approach (MLTCC). The top part of the window shows the search topic (A) and the most recently chosen prototype as well as all previously given feedback (B). It also contains a countdown timer indicating how much time is left for completing the current search (C). The results returned in each cycle are displayed in the central part of the window together with checkboxes that were used to mark relevant images in the evaluation (D). The bottom part contains four buttons: 'Start over' used to start a new search, 'Go to examples' which

allows to jump back to the screen with the example images, 'Go back' which can be used to undo the last cycle and 'Finish' which similarly to 'Start over' begins a new search but before that it also sends a report to the server (E). When a user places their cursor over a thumbnail image a larger version of the image is displayed. In order to provide feedback a user simply clicks on an example image, which refreshes the display using the user feedback. A number of example critiques are displayed below the main pane showing a number of example critiques (F). A user clicks on a critique to provide feedback which refreshed the display using the user feedback. For COPE using only MLT approaches the critiques are not visible. For COPE using only compound critiquing the users cannot click on images in the results pane to provide feedback but can mark images as being relevant (CC). In order to provide a comparison in a user study a text-based retrieval approach was also implemented (TXT). This interface is similar to the interface in Figure 2. It also contains a query input field and the 'Search!' button. A 'Next page' button is used to retrieve the next page of results. In this version users cannot provide feedback by clicking on images. Text search was implemented using Terrier (Ounis et al. 2006). In total four versions of COPE were used for evaluation (TXT, MLT, CC and MLTCC).

6. USER EVALUATION

6.1 Methodology

The evaluation was a within subjects evaluation. Each participant conducted one image retrieval topic using one of the four versions of COPE outlined. Of these versions TXT was considered to be an upper bound, as discussed (see Sections 1 and 2) earlier it represents the current state of the art. The order of system and topic were both rotated to avoid any ordering bias. Four ImageCLEF 2007 topics with different characteristics (Grubinger 2007; Grubinger and Clough 2007) were chosen to be used in the evaluation, these topics were:

- Topic number 2 - churches with more than two towers - a difficult semantic topic
- Topic number 14 - scenes of footballers in action - a difficult visual topic
- Topic number 30 - room with more than two beds - a medium-hard semantic topic
- Topic number 43 - sunset over water - a medium-hard visual topic

The topics gave a good coverage of probable search scenarios. The users were given a

Figure 2: Example COPE interface which allows more-like-this and compound critiquing feedback (MLTCC)

maximum of 10 minutes to complete each topic. For the systems which used only visual feedback, users could use the example images for each topic to retrieve initial sets of results. The number of results displayed on the screen in each cycle was limited to 32 which correspond to the configuration of other popular systems. For each participant their interaction with COPE was logged, the images they marked as relevant were stored and they also filled out a number of questionnaires at different stages of the experiment.

6.2 Results

12 users took part in the evaluation with average age of 22.75 (range=20-26). All of the users had some experience of dealing with images or videos, and all had used online multimedia retrieval systems such as YouTube or Flickr. All users also indicated that they use some systems with recommender services, e.g. Amazon.

6.2.1 Topic Performance

Table 2 shows the performance of the different versions of COPE in terms of precision, recall, how many relevant results are found using the different approaches, the average number of retrieval cycles (i.e. new query issued), the length of cycles and time to find relevant images. In terms of precision TXT more effective than any other approaches for 2 of the 4 search topics and in terms of recall it was more effective than 3 of the 4 approaches. A statistical multi-factorial ANOVA test was run on the results which indicated that there were differences for both the interface and topic variables ($p < 0.001$ in all cases). Post-hoc Tukey's range tests on individual factors only TXT was significantly different from other versions and the two visual topics from the semantic ones.

In general the trend was that when using the preference based approaches users had more retrieval cycles in comparison with TXT. Many of the preference based cycles were much shorter than the TXT approach. It appears that the users are seeing more of the collection and more quickly, thus demonstrating the possible applicability of the preference based approaches for exploratory search. These results are encouraging, for some topics the preference based approaches outperform the current state of the art and across all topics users using preference based approaches have more retrieval cycles and see more of the collection.

6.2.2 User Experience

In post topic questionnaires we solicited subjects' opinions on the versions of COPE that they had used. The following Likert 5-point scales and semantic differentials were used: "The results of the search were mostly relevant/irrelevant, expected/unexpected, complete/incomplete", "The search process was simple/complicated, restful/tiring, interesting/boring", "Deciding on feedback was easy/difficult, obvious/vague, flexible/constrained", "The interface was easy to learn/difficult to learn, easy to use/difficult to use, attractive/unappealing" and "Overall I think I performed well/poorly, efficiently/inefficiently". The feedback from the participants is shown in Table 3 and all results were compared using a one way ANOVA with system as a factor. System was a significant factor for relevant ($df(3,44)=2.953$ $p = 0.043$) (post hoc Tukey tests showed no pairwise differences), simple ($df(3,44)=4.304$ $p = 0.010$) (post hoc Tukey tests showed differences between TXT ($M=4.63$, $SD=0.389$) and CC ($M=3.75$, $SD=1.138$) and easy to learn ($df(3, 44)=2.955$ $p=0.043$) (post hoc Tukey tests showed differences between MLT ($M=4.58$, $SD=0.901$) and CC

Table 2: Figures for interactions and performance for each system and topic combination, best result in bold where appropriate

Topic	Approach	Avg. no cycles	Avg. cycle length	Time to first	Time to first rel.	Precision	Recall	%relevant displayed
43	TXT	6	34.02	34	34	0.534	0.628	0.829
	MLT	17	14.72	40	40	0.228	0.132	0.194
	CC	14.67	23.35	106	149	0.311	0.132	0.155
	MLTCC	16	22.86	64	64	0.247	0.194	0.248
14	TXT	8	27.08	66	66	0.495	0.392	0.558
	MLT	12	13.65	31	31	0.491	0.284	0.363
	CC	14.67	12.79	52	52	0.785	0.441	0.47
	MLTCC	15	19.55	73	73	0.49	0.255	0.412
30	TXT	12	25.3	75	75	0.12	0.16	0.507
	MLT	15	16.14	86	303	0.029	0.027	0.027
	CC	12.33	17.62	37	77	0.083	0.013	0.013
	MLTCC	11	16.46	27	-	0	0	0
2	TXT	6.67	33.07	73	206	0.038	0.153	0.264
	MLT	14.33	17.11	38	38	0.396	0.305	0.431
	CC	13.33	27.96	96	96	0.206	0.125	0.25
	MLTCC	12	26.81	74	74	0.34	0.236	0.347

(M=3.25, SD=1.128). There are some other trends that were noticed in the post topic feedback. Generally, CC got the lowest scores in most questions and MLT garnered a lot of positive responses. The TXT version got high scores for questions relating to perceived performance, although the results in Table 2 indicate that it might not have always resulted in the best performance.

Table 3: Average responses to post topic questionnaires, most positive response in bold. Significant differences are marked with *

Question		Approach			
		TXT	MLT	CC	MLTCC
Results	Relevant*	4.33	3.17	3.17	3.25
	Expected	4.25	3.58	3.17	3.5
	Complete	4	3.33	2.92	3.33
Search Process	Simple*	4.63	4.67	3.75	3.92
	Restful	3.92	4	3.08	4
	Interesting	3.5	4.08	3.58	3.75
Deciding on Feedback	Easy	2.92	2.17	3	2.67
	Obvious	3.83	3.42	3.08	3.75
	Flexible	3.27	3.09	3.17	3.83
Interface	Easy to learn*	4.08	4.58	3.25	4
	Easy to use	4.42	4.58	4	3.92
	Attractive	3.92	3.73	3.75	3.92
Perceived Performance	Well	4.17	4.17	3.58	3.75
	Efficient	4.17	3.92	3.25	3.58

After completing all of the topics and having used all versions of COPE the participants were asked to complete an exit questionnaire. The participants were asked, "Which of the systems did you...": "find easiest to use" (Ease of use), "find had the best results" (results) and "like the best" (Preference). The users were also given some space to provide any feedback that they felt may be useful. The median responses are shown in Table 4, a Friedman's analysis of variance by ranks was used to analyse the effect of the different interfaces on the rankings. Interface was found to have a

significant effect on ease of use ($\chi^2(3)=17.420$, $p=0.001$) with MLT having the best ranking. Post hoc Wilcoxon T comparisons (adjusted alpha = 0.008) showed significant differences between the MLT and CC systems ($z=3.111$, $p=0.002$). Interface types was also found to have a significant effect on results ($\chi^2(3)=17.483$, $p=0.001$), with TXT having the best ranking. Post hoc Wilcoxon T comparisons (adjusted alpha = 0.008) showed significant differences between the following systems TXT and CC ($z=2.952$, $p=0.003$), TXT and MLTCC ($z=2.835$, $p=0.005$), CC and MLT ($z=2.709$, $p=0.007$). For users' preference there were no significant differences. The responses to these three questions re-iterate the results from the post topic questionnaires, participants liked the MLT approach, did not like the CC approach as much and felt that TXT gave the best performance.

Table 4: Median and Interquartile Range for 1st and 3rd quartile for ranking of systems

	TXT	MLT	CC	MLTCC
Ease of use	2 (1-2.75)	1 (1-2)	3.5 (3-4)	3 (2-3.75)
Results	1 (1-1.75)	2 (1-3)	4 (3-4)	3 (2-3.75)
Preference	2.5 (2-3)	1 (1-2)	3.5 (2-4)	3 (1-3)

7. DISCUSSION AND CONCLUSIONS

We have presented a novel image retrieval system called COPE, which uses preference based feedback from users to allow them to query an image collection. This feedback is elicited as part of an extended dialogue which allows the user to quickly and efficiently browse and search the collection. There are a number of conclusions that can be drawn based on our results. While the text based system was considered to be an upper

bound, it was found that for some topics that the preference based image retrieval methods outperformed the text based system. This result is encouraging for a couple of reasons. First as the preference based image retrieval methods are slightly limited by the initial input image of which there is a limit of three. Secondly, users are engaged in more retrieval cycles, probably due to the simple feedback mechanism, meaning that possibly they may see more of the collection making the approach highly suitable for exploratory search or in circumstances where users have a broad or vague idea about what images they would like to retrieve. The participants indicated in their post topic questionnaires that the MLT only version of COPE had the best interface and the best search process. They also felt that the text based interface was best in terms of results; however, this may be that they found the results and interaction more familiar. In the exit questionnaire it is also shown that the users have a definite preference for the preference based image retrieval systems, in particular the MLT only version of COPE. This positive response of the participants is encouraging, especially as this is a new search paradigm and is being compared with a more familiar search paradigm. The results of the evaluation have highlighted the promise of this approach to alleviate the major problems that users have while searching for multimedia, presenting a potential work around to the semantic gap and other problems associated with image search.

8. REFERENCES

- Agrawal, R and Srikant, R (1994), 'Fast algorithms for mining association rules in large databases', *20th International Conference on VLDB* (Santiago, Chile), 487-99.
- Burke, R, Hammond, K, and Young, B (1996), 'Knowledge based navigation of complex information spaces', *AAAI National Conference on Artificial Intelligence* (Portland, OR), 462-68.
- Campbell, I (2000), 'Interactive evaluation of the ostensive model, using a new test-collection of images with multiple relevance assessments', *Information Retrieval*, 2 (1), 89-114.
- de Rooik, O, Snoek, CGM, and Worring, M (2008), 'MediaMill: fast and effective video search using the fork browser', *CIVR 2008*, 561-62.
- Flickner, M, et al. (1997), 'Query by image and video content: the QBIC System', *Intelligent Multimedia Information Retrieval*.
- Fogarty, J, et al. (2008), 'CueFlik: interactive concept learning in image search', *SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy), 29-38.
- Girgensohn, Andreas, et al. (2009), 'MediaGLOW: organizing photos in a graph based workspace', *13th International Conference on IUI*.
- Grubinger, M (2007), 'Analysis and Evaluation of Visual Systems Performance', (Victoria University).
- Grubinger, M and Clough, P (2007), 'On the creation of query topics for ImageCLEFPhoto', *Third Workshop on Image and Video Retrieval Evaluation* (Budapest, Hungary), 50-56.
- Guy, M and Tonkin, E (2006), 'Folksonomies: Tidying Up Tags?', *D-Lib Magazine*, 12 (1).
- Halvey, M. and Keane, M. T. (2007), 'Analysis of Online Video Search and Sharing', *ACM Conference on Hypertext and Hypermedia* (Manchester, UK), 217-26.
- Halvey, M., et al. (2009), 'ViGOR: A grouping oriented interface for search and retrieval in video libraries', *ACM/IEEE JCDL*, 87-96.
- Hauptmann, AG, et al. (2006), 'Extreme video retrieval: joint maximization of human and computer performance', *ACM Multimedia*, 385-94.
- Jaimes, A, et al. (2005), 'Multimedia information retrieval: What is it and why isn't anyone using it?', *MIR* (ACM Press), 3-8.
- Manjunath, B.S., Salembier, P, and Sikora, T (2002), *Introduction to MPEG-7: Multimedia Content Description Interface* (John Wiley & Sons Inc.).
- McCarthy, K, et al. (2004), 'On the Dynamic Generation of Compound Critiques in Conversational Recommender Systems', 176-84.
- (2005), 'Experiments in Dynamic Critiquing', *International Conference on Intelligent User Interfaces* 175-82.
- McGinty, L and Smyth, B (2002), 'Comparison Based Recommendation', *European Conference on Case-Based Reasoning*, 575-89.
- Ounis, Iadh, et al. (2006), 'Terrier: A high performance and scalable information retrieval platform', *SIGIR Workshop on Open Source Information Retrieval*.
- Ruthven, I. and Lalmas, M. (2003), 'A survey on the use of relevance feedback for information access systems', *The Knowledge Engineering Review*, 18 (2).
- Shimazu, H (2001), 'ExpertClerk: Navigating Shoppers' Buying Process with the Combination of Asking and Proposing', *International Joint Conference on Artificial Intelligence*, 1443-48.
- Shimazu, H, Shibata, A, and Nihei, K (2002), 'ExpertGuide: A conversational case-based reasoning tool for developing mentors in knowledge spaces', *Applied Intelligence*, 14 (1), 33-48.
- Urban, J and Jose, J. M. (2006), 'EGO: A personalized multimedia management and retrieval tool', *International Journal of Intelligent Systems*, 21 (7), 725-45.
- Villa, R., Gildea, N., and Jose, J. M. (2008), 'A faceted interface for multimedia search', *ACM SIGIR*, 775-76.
- Villa, R., et al. (2010), 'Can an intermediary collection help users search image databases without annotations?', *ACM/IEEE Annual Joint Conference on Digital Libraries*, 303-12.