IT UNIVERSITY OF COPENHAGEN

IT UNIVERSITY OF COPENHAGEN

DOCTORAL THESIS

Exquisitor: Interactive Learning for Multimedia

Author:
Omar Shahbaz KHAN

Supervisors:
Björn Þór JÓNSSON
Jan ZAHÁLKA

A thesis submitted in fulfillment of the requiremetns
for the degree of Doctor of Philosophy
in the
Data-Intensive Systems and Applications Group
Department of Computer Science

July, 2022

Abstract

Multimedia collections contain a wealth of information that can be used to gain insight into trends, performing investigations, finding media to represent concepts, and much more. Over the past decade, multimedia collections have seen tremendous growth, with technological advances allowing ever faster generation and sharing of multimedia data. Multimedia analytics is a research field that focuses on providing insight into large-scale multimedia collections. In this field, it has been stated that analysing such collections requires an interactive approach that combines the strengths of a human and a machine. The machine's objective is to present items relevant to the human's information needs, while the human may indicate their relevance, to improve the machine's future suggestions. To facilitate this process, an interactive learning approach capable of handling the scale of today's collections is required. While a preexisting approach can be scalable, it demands significant computational resources. In this thesis, a new interactive learning approach is proposed, called Exquisitor, which integrates high-dimensional indexing, incremental retrieval, and query optimisation policies into the interactive learning process, making it responsive, accurate, flexible, and scalable. Furthermore, the work emphasizes the need for better automated evaluation protocols, as existing protocols fail to capture different types of user interactions, making it reasonable to suspect whether or not interactive learning is suited for obtaining insight. New automated evaluation protocols are introduced in this work that analyse various user interaction strategies, to better evaluate the capabilities of interactive learning. While not eliminating the need for user testing, it allows for more detailed performance analysis of interactive learning approaches earlier in the developmental phase. Through extensive experiments, it shows that Exquisitor improves or maintains result quality, while drastically reducing response time and requirements for computational resources. In addition to the automated evaluation protocols, the approach has also been used in practice, through participation in live interactive search challenges. The research and development of Exquisitor has shown that interactive learning is efficient for gaining insight into large multimedia collections, establishing it as the new state of the art in large-scale interactive learning. By reducing requirements for computational resources, it opens up possibilities for future research on utilising these resources to introduce additional elements into the analytical process, such as concurrent classifiers, diversification of results, or dynamic combination of modalities.

Resumé

Multimedie kollektioner indeholder en stor mængde information, som kan benyttes til at få viden inden for trends, efterforskninger, finde medier til at repræsentere koncepter, m.m. Multimedie kollektioner har haft stor vækst i løbet af det sidste årti, da diverse teknologiske fremskridt har gjort det nemmere at generere og dele multimedie data. Multimedia Analytics er et forskningsfelt, der fokuserer på at analysere store multimedie kollektioner. For at opnå viden fra sådanne kollektioner, er der behov for en interaktiv fremgangsmåde, der kombinerer kompetencerne af menneske og maskine. I denne interaktive fremgangsmåde præsenterer maskinen relevante billeder og videoer for en persons behov, hvor personen har mulighed for at indikere medie objekternes relevans, hvilket maskinen benytter til at forbedre de næste relevante forslag. En interaktiv læringstilgang, som kan håndtere mængden af nutidens multimedie kollektioner, er nødvendig for at facilitere den ønskede fremgangsmåde. Én eksisterende metode er skalerbar og i stand til at behandle store kollektioner, men kræver mange ressourcer. I denne afhandling præsenteres Exquisitor, en ny interaktiv læringstilgang, der integrerer high-dimensional indexing, incremental retrieval, og query optimisation policies. Dette resulterer i en responsiv, akkurat, fleksibel og skalerbar interaktiv læringstilgang. Desuden fremhæves et behov for forbedringer i automatiserede evaluerings protokoller for interaktive læringstilgange. Eksisterende protokoller tager ikke holdning til hvordan forskellige personer interagerer med maskinen i en interaktiv læringstilgang. Dette kan lede til tvivl om fremgangsmåden kan analysere store multimedie kollektioner. Derfor introduceres nye automatiserede evaluerings protokoller, der analyserer forskellige måder en bruger kan interagere med maskinen. Disse ekskluderer ikke behovet for aktuelle brugertest, men fremmer forståelsen for effektiviteten af fremgangsmåden tidligere i udviklingsfasen. Omfattende eksperimenter har vist, at Exquisitor øger eller fastholder kvaliteten for at finde relevant data, samtidig med at reducere responstiden og mængden af ressourcer. Udover automatiseret evaluering er Exquisitor også blevet brugt i praksis, med deltagelse i live interaktive søge konkurrencer. Forskningen og udviklingen af Exquisitor har fastslået, at en interaktiv læringstilgang er effektiv for at opnå indblik i store multimedie kollektioner, og etablerer den som standarden inden for skalerbare interaktive læringstilgange. Reducering af ressourcerne åbner desuden op for ny forskning i hvordan de frigjorte ressourcer kan anvendes til at introducere nye elementer til at forbedre den interaktive process.

Contents

Acknowledgements

Over the course of my PhD journey I have had the fortunate opportunity to meet many amazing people that have helped shape this thesis, whether it be through direct collaboration, discussing research, or life in general.
I would like to thank:

My excellent supervisor **Björn Þór Jónsson**. From being your teaching assistant during my MSc to becoming your PhD student and going through this journey has all been an exciting and insightful experience. You have always had a welcoming atmosphere, that allowed me to grow and consult you about any idea at any time. The guidance you have provided throughout has been phenomenal, with a perfect balance of hands-off and hands-on. Even amidst all the stuff going on in the world, your busy schedule, from managing conferences, teaching obligations, supervising students, and the list could go on, I wholeheartedly appreciate that I could still approach you without hesitation. You have introduced me to many outstanding researchers from your network and shown me what defines a successful researcher, supervisor, and teacher. I am truly grateful to have had the privilege of working with you.

My co-supervisor **Jan Zahálka**. Starting from collaborations to you becoming my official co-supervisor has all been a pleasant experience. Your amazing ability to not only provide great feedback, but to always motivate and see the positives in any situation is something that I admire and have deeply appreciated in all our work, especially during these last few months.

Stevan Rudinac and **Marcel Worring**, for the collaborations and with organising my research stay abroad at University of Amsterdam. Your extensive knowledge of the Computer Vision and Multimedia Analytics field have had a significant impact on the work we have done. For all the great ideas, motivation, and guidance. **Dennis Koelma**, for your assistance with the access and usage of DAS-5, and sharing your knowledge of practical technologies. **Ujjwal Sharma**, for the collaboration and support during my stay in Amsterdam. To the **MultiX** group in University of Amsterdam, for the great work environment during my stay, especially given the circumstances at the time.

Aaron Duane, for the amazing friendship, support and motivation. The energy you bring is great, whether it be work related, having a “drink”, or a long random discussion.

Hanna Ragnarsdóttir, **Þórhildur Þorleiksdóttir**, and **Gylfi Þór Guðmundsson**, for providing an introduction to the original index implementations and the collaborations. To all the students that directly or indirectly impacted this work; **Liam** and **Mathias**, **Daniel** and **Felix**, **Kim** and **Alexandra**, **Emil** and **Thomas**.

To the **LSC** and **VBS** community for welcoming our, at the time, unorthodox system. **Jakub Lokoč** and his team for their initial feedback on our system when we visited Prague. **Klaus Schöffmann** and **Andreas Leibetseder**, for sharing ideas and making my visit to Klagenfurt very memorable. **Cathal Gurrin**, for the insights into these competitions.

I have to thank **Philippe Bonnet**, for the support during my MSc and when I first started the PhD, and most importantly I am thankful for you recommending me to be a teaching assistant for Björn.
Pınar Tözün, for the helpful comments and feedback during the midway evaluation as well as being thoughtful and understanding throughout.
Sebastian Büttrich, for helping me with the server and whenever I needed technical assistance. **Maria Sinziiana Astefanoaei**, for being so encouraging and supportive. **Ivan**, **Paul** and **Ehsan**, for sharing the office with me. To all the people in **DASYA**, I have seen this group grow from being only a few people to now filling an entire wing. Thank you for all the feedback during my presentations, but more importantly for all the support and friendly conversations over the years.

Lastly, the most important people in my life, my **parents** and my **brothers**. Your unconditional love and support helped me get through all the stressful periods, and always kept me motivated to achieve greater heights.

And to all the people who influenced this journey that I may have missed to mention here, thank you.

Almost forgot the special guest star that showed up midway through the PhD, “COVID-19”.

Chapter 1

Introduction

What do you think about when taking an image or a video these days? In the past, people have worried about how many images and videos they could take without running out of storage. Today this is a rare thought, not only because storage of all devices has significantly increased, but also because the majority of our multimedia items are stored in the cloud. With the storage restriction removed, people take images and videos at a greater pace than ever before, making personal collections much larger in size. We also upload media items to various websites and share them through social media platforms. This leads to massive multimedia collections that need to be managed in terms of storage and retrieval. Furthermore, these massive collections, ranging from millions to billions of items, contain a wealth of knowledge useful for understanding trends, performing investigations, discovering new concepts and finding media items that can represent them, and much more.

We have an expectation that machines are capable of understanding multimedia data better than in the past, as images and videos that are taken from our phones get automatically annotated. The reality is that pretrained models from deep neural networks are used to predict labels for images and videos, along with other computer vision techniques performed to extract more low-level features. While some may expect the machine to handle any sort of input, once we start going from words to phrases, most techniques start to struggle. This means that when users want to find items using elaborate labels and phrases, they have to either, simplify it into more basic labels, or start recalling when they took the image or video.

Additionally, with these recent technological advances, society today has become accustomed to a certain standard when it comes to interacting with applications. We expect rapid feedback from any application on computers, phones, or tablets, and we have a tendency to lose focus after waiting for 5 seconds or more [75, 84]. Much of this rapid feedback is due to advances in hardware and efficient data storage, that enables performing heavier computations on specialized hardware e.g. GPUs and FPGAs. Many issues related to growing collections may be resolved by adding more hardware. However, not all users may have access to such specialized hardware, nor the knowledge to optimally utilise it.

When users interact with a multimedia collection they typically approach it with a goal in mind. This goal can be broad, specific or somewhere in between. Consider the following two scenarios:

Scenario A A user wants to watch the penalty shootout from the EURO 2020 final between Italy and England. This is a specific incident in terms of time and place, making it easy to find through keywords. After the user finds the desired video the interactive session with the multimedia collection is over, or is it? The user's interest might now shift towards looking for triumphant moments in football involving England.

Scenario B A forensic analyst has been given a seized laptop containing a large multimedia collection, with the initial goal to find items containing criminal acts. The analyst might start by browsing the collection, but later realise that the goal may be too broad. Thus, they might focus on specific crimes and create a summary of items pertaining criminal elements, which they can later categorise into different priority levels.

Most common consumer applications allow users to interact with large multimedia collections through a search bar, or by supplying an example image or video, along with some filters that can be applied. When an example is provided, regular retrieval systems perform a comparison between the supplied example and the media items in the collection, to return the most similar ones. For scenario A, the user has to restart their search once their goal changes from the penalty shootout to triumphant moments in football, when using such retrieval systems. Analytical tools generally have more features to better support the actions the analyst needs, but do not support the type of flow between exploration and search present in scenario B. Even if a system did support such a flow between exploration and search, an issue still arises with regards to scale. When it comes to massive collections there are more advanced techniques that support scalable search than scalable exploration. However, even some of these techniques have a tendency to rely on more powerful hardware. What is missing in today's environment of multimedia search and exploration tools is a scalable interactive approach that is able to adapt towards the user's goal, regardless of whether it is oriented towards exploration or search, while keeping the cost of computational resources low.

During retrieval with example images or videos, the machine uses representations of their content to figure out which items in the collection are similar. Many features can be derived from a multimedia item and be part of the representation, leading to high-dimensional data. Using high-dimensional data representations to compute similarity between the examples and all items in a collection can be costly, especially for larger collections. To facilitate scalable retrieval for multimedia collections, high-dimensional indexing is needed, where the representations of multimedia items are split into smaller areas, which reduces overall computation. Multimedia Analytics is a research field that focuses on extracting information from multimedia collections to obtain various insights. In this field, interactive approaches that allow a human to work together with a machine have been suggested as being capable of obtaining insights from large multimedia collections. For a human and machine to work together in such a setting, a set of requirements need to be met. These requirements focus on the ability towards finding relevant items for a given information need and for the performance at scale [113]. Interactive learning is an approach where human and machine work together in order to support better retrieval. In its simplest form, a screen of suggested items is provided by the machine, and the human user determines their relevance towards the goal or insight they want to obtain. The machine uses the feedback provided to update its underlying model and presents a new set of suggestions for the user to go through. This is an iterative process which stops when the user decides. With such an approach the user has more control over the direction the interactive session is going into.

This thesis presents Exquisitor, a scalable interactive multimodal learning approach that addresses the need for solving analytical tasks with modest resources. Exquisitor combines and extends state-of-the-art approaches in interactive learning, storage, and retrieval to facilitate interactive sessions capable of solving exploration and search tasks. In addition, the thesis expands on the current evaluation efforts of such systems, to better understand the influence of user behavior earlier in the development phase. As multimedia collections consist of different media items (images and videos), with multiple layers of information that can be extracted, such as objects from images or actions from videos, the impact of using multiple information sources is analysed. Through extensive experiments, Exquisitor is shown to be capable of interacting with collections of over 100 million items, with higher accuracy, lower response time, and lesser resource demands, than state-of-the-art approaches. These results establish Exquisitor as the new state of the art.

The remainder of the thesis is structured as follows: Chapter 2 covers the background of multimedia and interactive learning. Chapter 3 presents the articles that make up the foundation of the Exquisitor approach. Chapter 4 highlights and accounts for the demonstrations and workshops where the Exquisitor client application has participated in. Chapter 5 summarises the work described in the thesis and addresses future research prospects.

Chapter 2

Background

“A picture is worth a thousand words”, is a well-known saying that implies that humans are able to make many observations from just looking at an image. These observations can relate to colors, shapes, objects, or *concepts* based on the individual’s knowledge, such as types of cars and distinction between pets and wild animals. This is in stark contrast to how a machine *processes* an image. For the machine to determine any sort of content within the image, it needs to have meaningful *representation.* The attributes of a representation, called *features*, are numerical values describing various parts of the contents in an image. These features can be low-level, such as intensities of pixels to determine shapes, or high-level, such as probabilities for concepts or objects being present in the image. Note that unlike humans, the concepts or objects a machine can derive are finite. Aside from the visual content of an image, the machine also has access to metadata. This can be technical data relating to the capturing device or an external device, or it can be annotations provided by a human in the form of descriptions or tags. The different types of features from the multimedia content, is referred to as a *modality.* Thus, metadata features may relate to the textual modality, while content features relate to the visual modalities such as concepts or actions, and for videos features relating to audio is another modality. Recall scenario B, where the forensic analyst wants to find images and videos from a seized laptop relating to crimes. When going through the media items, the analyst will know when an item includes a suspect or people related to a suspect, while a machine’s representation will at most lead to concepts such as “woman and man talking” or “man in a sports car”. Context is rarely part of the multimedia representation, as it is quite difficult to obtain through objective means. To make the machine find items with specific context, *interaction* with a human is needed, as they can tell the machine which items are relevant for their context. Facilitating such an interactive approach, requires the representations from the machine to be transparent towards the user, so they can comprehend why certain interactions lead to a specific outcome. Section 2.1 goes into details of different multimedia representations and their impact on various tasks.

The machine processes an image by deriving one or more representations from it. The representations of a multimedia item can be of varying sizes and are often of a high-dimensional nature, such as feature vectors detailing the orientation, size, and shape, for a set of pixels, or probabilities of concepts from the machine's concept dictionary. These representations are used to find relevant items when a user interacts with the collection. For instance, to find items related to crimes, the analyst can start by browsing the collection through an explorer, or query a search engine with text or an example image containing crime related concepts. To determine the similarity between the examples and the media items in the collection, a similarity score is computed using the feature values of their representations against the examples. This similarity score is used to sort the collection, from which the top items are presented to the analyst. With such representations, computing the similarity for all items in a collection can be time consuming, especially if more than one representation is involved and if the collection is very large. Typically, to reduce computation and retrieval time when searching in a database, indexes are used to organise the data into smaller areas for quicker access. For the high-dimensional representations of multimedia data, specialised high-dimensional indexes are required, which focus on storing the multimedia representations for scalable similarity based operations. Section 2.2 covers multimedia retrieval using high-dimensional indexing.

A human can formulate their task to the machine in multiple ways, such as a textual description, keywords, or when it is difficult to find words to describe the task, they may supply example images. The machine attempts to either find exact matching items in the collection or similar items. If the task is to find multiple relevant items to the supplied images, finding visually similar items may not always lead to the correct items. Assume the forensic analyst is searching for crimes related to assault and supply an image of people fighting. The machine's representations of the media item heavily determine the results for this scenario. If the machine's representations are capable of extracting the concept of "fighting", the output may be images and videos of boxing and UFC events, along with actual items containing assault. In most systems, the analyst will now have to either browse through the results, or supply different examples and hope for better results. Alternatively, a far better option is to guide the machine towards the relevant concepts and context, by stating which items from the results are relevant and which are not. In this case, the boxing and UFC items are not relevant, while the items with assaults are. Now, the machine can use this feedback to train a classifier and get better items fitting the context and concepts. If the output still contains irrelevant items, the user can continue providing feedback. This sort of interactive approach is known as Interactive Learning and is not only ideal for assisting the machine in understanding the task of the human, but also for the human to explore the collection. Section 2.3 takes a look at how interactive learning approaches have been applied to multimedia collections.

Evaluation relating to the performance of retrieval systems often focuses on the quality of the results and the response time of operations, using automated benchmarks. For interactive learning approaches, the user is involved far more than just providing an initial query, which is difficult to reflect using automated evaluation efforts. User studies are ideal for interactive approaches, but acquiring general users can be difficult, let alone actual domain experts such as forensic analysts. Performing user studies on an early development system is not ideal, as there are typically a limited number of available users, which means they have to engage with the system on multiple occasions in the development phase, thus introducing bias. Furthermore, early stage systems that want to simply test the algorithmic performance, may not have considered the interface design. This may influence the user's behavior in the tests and can unintentionally mislead the results. If the developers are fully aware of this, they can use the opportunity to test both functionality and the related interface, but this increases the workload of developers. Thus, automated evaluations have their merits during development and provide foundation for an approach's capabilities. To properly evaluate an interactive learning approach, using automated evaluation early on and proper user studies at later stages is ideal. Section 2.4 reviews automated evaluation efforts and evaluation with real user sessions for interactive learning.

2.1 Multimedia Representation

A great amount of information is captured when an image or video is taken, from technical details about the capturing device, to the visual information of its content, along with supplementary visual information extracted by a machine, and textual information such as description or tags provided by a human. The information residing in a singular media item is useful for obtaining knowledge for many purposes. For instance, a forensic analyst may be interested in learning about the camera device, or if the items contain descriptions or tags of interest, or to learn more about the visual surroundings related to a suspect. While an image or video may provide valid evidence for a task, associating the information between items in a multimedia collection through one or more modalities, is beneficial for reinforcing known knowledge, or expanding ones knowledge. Thus, it is important to know how different types of modalities are represented by the machine, to determine their suitability towards different tasks.

In general the resulting file from taking an image contains the image data, along with EXIF data, which relates to the physical camera device, its settings, and time and date. If the device has a GPS then there is a high possibility of geolocation data also being part of the EXIF data. EXIF can further contain copyright information, image description, user information and more, but the majority of these are optional and regular people do not bother adding them. Some users that are knowledgeable with EXIF data, may see it as containing too much personal information or that it increases the size of an image, and decide to remove it. Photographers may remove it to not expose the settings of the device they used. Even so, EXIF data is the initial metadata for majority of images available on computers. While not the most descriptive information, it is still enough to find items, e.g. using geolocation if a user wants to find images captured in a specific city, or the device name if a user is looking for images captured by their smartphone and not their handheld camera. Although it is possible to add image descriptions to an image file, textual descriptions of images are usually provided when people upload their images to a social media platform or sharing site. They may also add tags related to content or the context of the image to get more exposure, and add it to an album on the sharing site or their device. Without processing the image or the metadata for additional information, this is the available data, which is mostly useful for finding exact images or within specific ranges. In the case of the forensic analyst attempting to find crime related media items, the metadata can at most be useful, if the descriptions or tags contain information related to the suspect, such as name or nickname, or if the analyst knows that certain crimes took place in specific areas. If the analyst wishes to focus on the contents of media items depicting crimes, EXIF data from the items is unlikely to contain much information about the contents.

When it comes to the contents of a media item, many different representations have been used for the visual elements in multimedia items over the years. Early on, the representations focused on the physical characteristics of an image, using information from the pixels. With colored images, the intensity of pixels from different color spaces, such as RGB, can be used to define color histograms or correlograms [31, 42, 97], which are useful for determining the presence of dominant colors or spectrum of colors. Aside from colors, the information from pixel intensities can determine specific textures, as well as outline shapes [31, 73], and be used for facial recognition [5, 115]. While these representations are useful, they are heavily linked with the item in terms of position, size, rotation, etc. As such, attempting to find images that contain a book, using an example image where a "book" is on the left side, will result in items with a book on the left side. With local invariant features [101] such as the popular scale-invariant features (SIFT) [69, 70] this issue can be avoided. SIFT features represent an image with multitudes of feature vectors containing information about the various low-level details. These features focus on the low-level characteristics of an image, and do not necessarily capture the concepts within it.

To make the machine better at grasping semantic concepts, machine learning techniques can be used to train various models that classify the contents of an image [57]. The classification is based on annotated data used for training, where annotations are provided by humans. For a training item, either all highly visible concepts seen in an image are annotated or the most dominant concepts. With multiple annotations for an image, the location of the concepts can also be added in the form of bounding boxes. Initially, the classification for these concepts, through machine learning approaches, were low in terms of accuracy and the number of concepts. However, with the emergence of deep learning, it has become apparent that learning features using deep convolutional neural networks (DCNN) is significantly better for extracting semantic concepts [59, 98]. The primary drawbacks of DCNN's are the number of annotated items they need for training, the time it takes to train the model, and the finite number of output labels, but even then the gain is so substantial compared to prior approaches that it is seen as an acceptable trade-off. This is a highly active research field, with many new neural network architectures being presented each year that aim to increase accuracy, reduce training time, optimising hyperparameters and more [30, 95]. To circumvent the training, pre-trained models based on different datasets, such as ImageNet, are available to use out-of-the-box for extracting features, making the use of deep net technology extremely appealing. On top of semantic concepts, actions [38], scenes [117], fashion/clothing [66], and other specific categories of labels can be extracted with deep learning models. The representation from DCNN is a vector containing the values of the output layer in the neural network. These vectors are dense and relate more to the machine's understanding of what is in the image. Alternatively, a softmax function can be used on the output layer to obtain probabilities for the presence of all the labels within an image, which are sparse and comprehensible for a human.

There are many ways to represent the textual data. In the most basic form, a key-value format can be used, but this primarily leads to heavily search oriented queries that rely on the user having extensive knowledge of the item(s) they are interested in. Instead of a relational format, the text from fields such as descriptions and tags can be represented with a vector space model, which can be used for similarity-based search. This representation can be fairly basic, such as using a bag of words [23], where all words from a field such as description are used to define a vocabulary. The representation for an image can then be the count or frequency of the words from the vocabulary. They can also be represented with a vector of term weights using TF-IDF [83], which determines the importance of words from an image file's text field such as description, based on their presence in that image and the rest of the collection. This can also be used to remove words with low scores, which are determined to be of low importance [67]. Alternatively, neural networks can be used to train a word association model such as Word2Vec [22, 44], that can output aggregated word vectors of the textual data. These word vectors can be used as is, but they tend to be fairly large and incomprehensible to a human. To alleviate this process, topic modelling is performed, where a set of relevant topics is determined, and the similarity score between the words from an items text and the topics is calculated and used as the representation [46]. With great progress in natural language processing and deep learning, recurrent neural networks can be used to extract long short-term memory (LSTM) embeddings (feature vectors) [105]. These representations take into account the relation between the words within a sentence and are better suited for representing sentences. Ultimately, the vector space models are well suited for similarity queries, that allow the user to search with fewer constraints.

With all the different types of multimedia representations, it is important to know their benefits and drawbacks. For instance, the visual modality representation of SIFT features is well suited for copy detection tasks [62], as the numerical features indicate the specific details forming an image. They are ill-suited for tasks focusing on finding concepts within images. Instead, using feature vectors derived from a DCNN will be a better fit. Similarly, for the textual modality, if the user wants to find items related to a tag, a key-value representation will be enough, whereas if they want to match phrases from descriptions, a vector space model may be better. Once the type of interactions and tasks for a retrieval system are clear, then appropriate representations from different modalities can be chosen for the multimedia items in the collection.

With multiple representations for items, there is the option of combining their information to improve the analysis and retrieval processes. To combine and use multiple modalities certain choices regarding *fusion* need to be made [7]. There are two types of fusion that can be considered, early and late fusion. Early fusion implies that the data representation is of a joint nature between the used modalities, typically by concatenating the different representations [96]. Many deep learning approaches have also been used in similar fashion to train models using multiple input modalities leading to a combined representation through some form of concatenation/fusion layer [10, 18, 41, 78]. This can also be considered as mid fusion since it is essentially using the individual modality representations initially [50]. Late fusion typically retrieves items from each modality and then combines their result sets, before presenting them to the user. The combined result set can be based on rank aggregation [65] or the modalities combined scores [103, 116]. The benefit of early fusion is less storage requirements as only one representation is being used, though it can be less transparent and more complex to perform computations on. Late fusion does require additional storage as it uses multiple representations that need their own storage and retrieval structure, but it opens up for more actions to take, such as determining the influence of certain modalities.

2.2 Multimedia Retrieval and Indexing

For very small collections, it may be possible for a human to go through each item and find items that are related to their task. For larger collections, however, it is infeasible that a human can process them entirely on their own, and therefore we rely on a machine by using its capabilities for efficiently storing and processing larger collections. For the machine to facilitate such retrieval, it needs to store the multimedia representations in data structures that are designed for efficient access. There are multiple ways to perform a search on a multimedia collection. The most common forms are query by text, applying filters (faceted search), and query by example, while other forms such as query by sketch or query by concept/object location are less common [2, 39]. The results of a search can differ depending on the objective, such as searching for exact items containing the query information or searching for similar items.

Search for images and videos often requires methods involving text based search, either through phrases or keyword terms. For this type of retrieval, the item's metadata along with auto-captions from the visual modality, can be used to build an inverted index to map all the words from the representations to the items containing them [17, 107]. These are useful for finding exact matches, as well as items partially containing the supplied query text. Another way to perform query by text is to use the vector space models of the textual modality from LDA topics, or from deep net models such as LSTM's, or Word2Vec. These representations are used to determine which items are most similar to the query text. This is also how query by example is performed, using representations such as DCNN models or SIFT features. Query by example is convenient when the user is unable to determine the right words to describe their item(s) of interest, or is looking for items containing the exact same information. To find the most similar item(s) a typical search strategy is to use the well known k-Nearest Neighbor (k-NN) algorithm [11].

When the contents of multimedia representations are high-dimensional vector space models, high-dimensional indexing is used [13]. These indexes focus on splitting the data into smaller areas that are represented by either a combined representation of the underlying items, or a selected representative item. To ensure that items are found within the smaller areas of the index, an item's representation can be duplicated to multiple areas of the index. High-dimensional indexing is optimized for similarity search [12, 14], where Approximate Nearest Neighbor (ANN) search is frequently used, usually as a variation of k-NN [9, 19, 25, 60]. Unlike k-NN, where the entire collection need be processed, ANN focuses on a pre-determined number of areas in the index neighboring the query, which significantly reduces computational load. A downside to ANN is that relevant items may not be part of the checked areas of the index. In such cases, measures can be taken to expand the search space by checking more areas [40, 89], which may be easy or difficult to do depending on the index structure.

The data structure of a high-dimensional index can be hash-based. The intention behind hash-based approaches is to use a hash function that transform the representation of an item into a unique fixed-size value. In general, hash-based indexes store the data into buckets. When finding exact matches, the goal is to locate the best matching bucket and search through it. To improve data distribution in non high-dimensional hash-based indexes that focus on finding exact items, use hash functions where close data representations have a larger distance. For high-dimensional data, where the focus is on finding similar items, having large distances between close items is not appropriate, as close items will end up in different buckets. Hash-based indexes for high-dimensional data, such as the popular Locality Sensitive Hashing (LSH) approach, attempts to do the opposite, by using hash functions where close representations also have close hash representations. An LSH index uses multiple hash functions to get multiple hash representations of an item, which are stored in a number of tables and buckets [3, 27]. During retrieval, the example query is transformed using the hash functions and an appropriate bucket is found from each table, which is then searched to find the k most relevant items. This approach leans heavily towards search, as the process focuses on looking into items from the most appropriate bucket of each table. This can make it difficult to move towards exploring a collection in an interactive approach. Multi-probe LSH variants allow extracting items from more buckets, by also checking a number of surrounding buckets [71, 72, 104].

Vector quantization approaches are another popular way to construct high-dimensional indexes. They typically project the multimedia representation to a line using a distance measure and store it in a classic B$^+$-Tree [32]. A modified version of this is the Nearest Vector Tree (NV-Tree) that constructs a tree from repeated projections to arbitrary lines [61]. This has been proven to be highly scalable, both in terms of growing collections and increasing number of dimensions of representations [63]. These are appropriate for ANN search by projecting the query to the arbitrary lines and finding k relevant items. Product Quantization is another vector quantization approach that first compresses the representations and then places the data into clusters [34, 45]. By compressing the representations, it reduces computations between items. Cluster Pruning is an approach that forms a multi-level index of clusters, where cluster representatives are arbitrarily selected like the first step of k-means, then the tree is built bottom up. The algorithm focuses on creating balanced clusters and during search selects items from b clusters, which can be altered at runtime. This leads to a trade-off between search speed and result quality [21]. Extended Cluster Pruning (eCP) is a variation that targets cluster sizes that fit within a single disk I/O in case the index is too large for main memory [36, 76]. For the cluster based approaches, it is simple to perform ANN search by selecting a maximum number of clusters to go through.

All of these indexes use the query example as a point to find the nearest neighbor around. However, with an interactive approach the user may supply multiple items, in which case the point is a mean of their feature vector. This can easily lead to unreliable results. Alternatively a classifier can be used to to define a decision boundary for relevant and non relevant items, such as the linear SVM [24]. In such a case the items farthest from the decision boundary in the relevant direction are the desired items. This requires the query process to find the farthest neighbor to a plane, rather than a point. There is research in approximate farthest neighbor to point queries [79], whereas farthest neighbors to a plane are not prevalent in high-dimensional indexing.

2.3 Interactive Learning

Interactive learning is a human-in-the-loop approach where a user is presented with items that they need to provide feedback on. The feedback is used to train an interactive classifier, that is used to retrieve a new set of items for the user to judge, concluding one interaction round. This feedback loop continues until the user decides to stop, either because they found their desired item(s), or determined that no relevant items are present in the collection. The way a user provides feedback is by labeling items as positive or negative. The systems may ask to label all presented items, a fixed number, or leave it up to the user [58, 87, 100, 118]. Studies have also shown that only providing negative feedback is also a possibility [106, 109]. While the machine may be capable of processing a large number of items, humans get overwhelmed when too many items are presented to them. As such, interactive learning approaches focus on presenting a small subset of the items requiring feedback, instead of a ranked list of the entire collection.

Interactive learning consists of two forms, active learning (AL) and user relevance feedback (URF). In active learning the feedback process is not part of the retrieval process. Instead, the intention is to spend a short number of interaction rounds to define an efficitive classifier towards the desired information need [35, 43, 93]. This is done by presenting the user with items that the underlying interactive classifier is unsure about, typically data points near the decision boundary, usually for a pre-determined number of feedback rounds, or until a threshold is passed using a stopping strategy [64, 74], whereafter the classifier is used to retrieve the most confident items. Given this approach of teaching the model first and then using it for retrieval, AL approaches attempt to focus on reducing the feedback time or labelling cost by locating high-value items that heavily improve the model [33, 102]. AL approaches can be beneficial for tasks that include notions which the machine may not contain in its representations. However, for longer-running tasks where the items of interest may change over time, AL will need to restart the learning process every time additional information to a task is added. AL has been shown to be better for optimising classifiers or models, and has been getting more attention recently with deep learning methods. As these methods require significant annotated data and training time, AL can be used to provide critical feedback during the training to shift the model in the desired direction [1, 33, 68, 110]. The main drawback of AL is primarily deciding when to stop the feedback process, and research is continuously being done to determine optimal stopping strategies [64, 74].

User relevance feedback has been used in content based retrieval from the very beginning [29, 86, 118]. URF focuses purely on the retrieval, where it aims to present the classifier's most confident items after any given round. This allows the user to continuously influence the retrieval process throughout the interactive session. Although user relevance feedback aims to show the most confident items, diversifying the results by also including some items farther from the decision boundary of the classifier is a possibility [28]. A variation of URF is pseudo-relevance feedback (PRF), where the user is removed, and instead the top most relevant items for an initial query is used as positive examples to update the classifier, and then get the actual results [15, 47, 108, 109]. Notice that nothing prevents AL and URF from working in the same system [114], but the choice of method to use still affects the workflow.

Latency is of great importance in interactive learning systems, to ensure that the user does not lose focus. While advanced classifiers, using deep learning or other machine learning approaches, may result in greater quality, the sheer amount of time they take to train makes them unsuitable for an interactive approach. Take for instance the forensic analyst example, where the analyst may only have a few hours to find crime-related media items. Using an interactive approach that takes 1 minute per interaction or 1 second per interaction can make a major difference. There are two reasons for the high training time, the first being the architecture of the classifier, but the greater factor being a large number of training items, as without it the quality drops significantly. Therefore, it is common to use these approaches to preprocess the items to extract features for a representation, and for retrieval k-NN or the popular support vector machine (SVM) is used [20, 92, 99]. The SVM is an appropriate choice for an interactive classifier as it is efficient and requires only a few training examples. Another classifier that has been shown to be valuable is the Self-Organizing-Map (SOM) [58]. Other approaches use nearest-neighbor queries with weights that are optimised based on the feedback [29]. In comparison to these SVM is better qualified for exploration tasks as well as search oriented ones.

Prior to the work presented in this thesis, the state-of-the-art scalable interactive learning approach was Blackthorn. Blackthorn is able to interact with a 100 million item collection with an average latency of 1.2 seconds per interaction round. It uses URF with a Linear SVM as its classifier, and uses a highly efficient compression technique that results in 99% reduction of feature vector size, allowing the data to be in memory. If multiple modalities are used, it performs late modality fusion through rank aggregation, but only on a fixed number of candidates from each modality [111]. While Blackthorn adheres to the requirements set for interactive human and machine approaches [113], a drawback is that it achieves its impressive performance by using 16 CPU-cores, and if the collections grow, more CPU-cores are needed to maintain the same response time. For a truly scalable interactive learning approach, there is a need for more efficient data structures [48].

2.4 Evaluating Interactive Learning

Evaluation of interactive learning approaches typically focuses on the precision and recall to determine the quality of the approach. While focusing on the quality may show that the interactive approach is able to find relevant items for a user's needs, response time is equally important with growing multimedia collections, as highlighted in the previous section, especially with society being accustomed to instantaneous responses from interactive applications. High quality is always desired, but if it takes too long, the user may lose interest or run out of time if the task is time sensitive. Latency is also important to consider when encountering tasks with incremental descriptions, meaning more information is added to their description or replaced over time as additional knowledge is gained.

The actual experiments to evaluate an interactive learning approach can be done with automated evaluation protocols or through user studies. While both methods have their benefits, automated evaluation protocols can be performed at any given time with metric outputs, whereas user studies need to be scheduled and properly analysed to provide both quantitative and qualitative data. Furthermore, with interactive learning approaches, there are certain parameters that are user dependent, such as number of items presented, and number of items labeled as positive and negative. These can be preset for an automated evaluation protocol, and be enforced in a user study through suggestion or implementation [86, 100]. For early stage systems, automated approaches are useful to indicate how viable an approach may be, while user studies on a mid to late development stage system are far better than automated evaluation, as they can highlight interaction patterns which may not have been accounted for in the original design.

Automated evaluation protocols are fairly similar to regular retrieval benchmarks, where a task description is used to query the underlying data structure to get the top results. The difference here is that these need to reflect interactive learning, which means they need to represent a user providing feedback to a small suggestion set, throughout an interactive session. The interactive session is typically limited to a number of rounds or until all relevant items from the ground truth are found. To reflect real users, these protocols use artificial users which are assigned a task with a corresponding ground truth. For each interaction round, the artificial user uses the ground truth to label positive examples [26, 100], with some evaluations adding arbitrary negatives from the collection as well [80, 85, 112]. The value of automated evaluation protocols is to get an indication of the interactive approach's performance during the developmental phase. Most evaluation protocols simulate perfect users with finite tasks. Human users are not perfect and are prone to make mistakes, which can be reflected by allowing the artificial users to make mistakes [114]. Additionally, each human user will interact differently towards the interactive system, which can be due to their knowledge and experience of the system or collection, or their understanding of the goal/task. Thus, the artificial users' behavior with regards to labeling items needs to better reflect a real user. Furthermore, to better evaluate the capabilities of an interactive learning approach for exploration and search tasks, tasks with descriptions changing over time and with objectives differing from finding one or more items, are required.

While it may be difficult to set up user studies for early development systems, acquiring feedback for an interactive approach in a real setting is beneficial, as an approach may do well in the automated evaluation, but performing an analytical task in a real setting will help uncover whether or not the approach works as intended. Therefore, simply demonstrating the approach to an audience, or asking someone to perform or state a task, is an opportunity to get valuable feedback of the approach. Interactive search challenges such as the Lifelog Search Challenge [37] and Video Browser Showdown [91] are live workshops where multiple interactive retrieval systems compete to solve various search oriented tasks. The tasks vary from finding one relevant item to finding as many as possible, and are either presented with a video segment, a fixed textual description, or an incremental textual description where more details are provided over time. These venues are great testing grounds for interactive retrieval systems, however, they are collection specific with tasks primarily leaning towards search. Typically, systems that include multiple forms of representations and retrieval methods tend to do well. While interactive learning approaches are capable of solving search tasks, they tend to start from an explorative perspective which may not hone in on the relevant items as quickly as a pure search approach. Nevertheless, these venues are still beneficial for evaluating the adaptability and real-world performance of interactive learning approaches for a specific group of tasks.

Chapter 3

The Exquisitor Approach

This chapter presents Exquisitor, a new interactive learning approach for large scale multimedia, along with new ways to evaluate such interactive approaches. Exquisitor combines user relevance feedback with high-dimensional indexing, incremental retrieval, and query optimisation policies, to achieve a scalable interactive learning approach that reduces computing resources, while maintaining or improving the quality over other such approaches. This chapter consists of 4 articles that covers the foundation of the Exquisitor approach. Chapter 4 details Exquisitor as a system that is demonstrated and used in interactive live search challenges.

Recall that the large-scale interactive learning approach Blackthorn manages to interact with a collection of 100 million items, with an average response time of 1.2 seconds on 16 CPU-cores. If only a single CPU core is used, the average response time increases to roughly 5 seconds, indicating the reliance on available computing resources. It is inevitable that a scalable approach will rely on available resources to a certain degree with large multimedia collections, but this can be alleviated through high-dimensional indexing. While any index may improve the response time and reduce resource requirements, it is important that it maintains similar quality and allows solving fluctuating tasks, such as scenarios A and B from the introduction (Section 1). Therefore, the chosen high-dimensional index needs to satisfy a set of requirements to achieve the desired efficiency.

Section 3.1 presents the article entitled **Interactive Learning for Multimedia at Large**, published in the proceedings of the 42nd European Conference on Information Retrieval (ECIR 2020) [54]. This article introduces the requirements for a large scale interactive learning approach together with high-dimensional indexing, and proposes the initial version of Exquisitor as the approch that best satisfies these requirements. Exquisitor uses the foundation of Blackthorn; user relevance feedback with a linear SVM classifier, along with a comprehensible compressed representation. The high-dimensional index that is deemed most fitting for interactive learning is the extended Cluster Pruning (eCP). eCP is an index that has primarily been used for nearest-neighbor search to a point, which is not how relevant items are found using decision boundaries of the SVM's hyperplane. To find the most relevant items for an SVM's hyperplane, farthest-neighbor queries to a plane need to be supported. Exquisitor uses a modified version of the eCP that supports k-FN queries to a plane and integrates the compressed representations of the multimedia items. To evaluate the capabilities of the initial Exquisitor approach, with regards to classifier adaptability and performance at scale, two evaluation protocols are used for the ImageNet and YFCC100M collections, which are image collections consisting of 14M and 100M items respectively. The results of this evaluation show that the linear SVM is capable of adapting towards new/unknown knowledge, and the performance of the eCP index achieves similar or improved quality. In terms of response time and lowering resource requirements for collections at YFCC100M scale, the approach achieves an average response time of 0.3 seconds using only on a single CPU-core, which is 0.9 seconds faster and with 16x less resources than Blackthorn. While these results do establish the initial version of Exquisitor as the new state of the art, the evaluations used do not consider human user behavior, when it comes to interactions with interactive learning systems.

In the evaluation protocols used for interactive learning, the items are labeled based on the ground truth set, where anything in it is positive and everything else is negative. In a real interactive learning session, the number of items one user perceives as positive and negative may be different to what another user perceives. Furthermore, a real user will also label non-relevant items as positive to steer the model into the desired direction of relevant items. For more search-oriented tasks, users typically have the option to apply filters to reduce number of items being considered. Filters are set based on a user's knowledge of the collection, where a novice user may apply filters that do not significantly reduce the scope, an experienced user may know of filters that fit better for a given task. It is important to understand the effects of the different ways a user can interact with an interactive learning approach, to enhance existing features or reveal the need for additional features.

Section 3.2 presents the article entitled **Impact of Interaction Strategies on User Relevance Feedback**, published in the proceedings of the International Conference on Multimedia Retrieval (ICMR 2021) [53]. This work defines a set of labeling and filtering strategies, with the former based on observations from real user sessions and existing protocols, and the latter based on different levels of knowledge of a collection. To evaluate these strategies, new evaluation protocols are defined where the tasks focus either on finding one relevant item or all relevant items from the ground truth. The artificial users in these protocols label positive examples based on a distance from an item to the ground truth items. The analysis of this work refutes the common assumption of more training examples always being beneficial. It also indicates that arbitrary selection of negatives from the collection is valid for tasks with many similar non-relevant items, and shows that aggressive filtering by users with lesser knowledge of the collection can lead to excluding relevant items. There are still aspects of interactions that these new evaluation protocols do not consider, but they shed light on the fact that the human user is a major factor in these approaches, both in terms of effecting quality and time to complete a task.

For large collections, users are not presented with all relevant items for their query, but a smaller subset of the top relevant items, as to avoid overwhelming them and to reduce time spent on considering which item to label what. If the user has a search-oriented task in mind, they may wish to apply constraints in the form of filters, to narrow the search space. One of the major reasons behind Exquisitor's performance is the high-dimensional index. A common trait in approximate high-dimensional indexing is to split the data into smaller areas and only process a limited number of those areas during retrieval, constituting a responsive approach. To maintain a quick response time, filters are only applied on the items from the selected areas. With a limited number of areas, applying filters may lead to no suggestions being found. Returning no suggestions is acceptable if there are no items in the collection that pass the applied filters, but for approximate high-dimensional indexes this is not ensured since only a subset of the collection is considered. Ensuring that all items are checked when aggressive filters are applied, requires the ability to expand the search space; considering items from additional smaller areas, when the set of suggestions is small or empty. The expansion needs to be done through increments, to avoid unnecessarily processing a large number of items, but this incremental retrieval approach can increase response time if it has to expand the scope multiple times. To alleviate multiple expansions, query optimisation policies can be used based on the information of an area's items, to establish whether or not it is worth processing. This way, only areas that contain one or more items passing the filters will be processed.

Section 3.3 presents a journal article entitled **Exquisitor: Responsive, Accurate, Flexible and Scalable Interactive Learning for Multimedia**, which has been submitted to IEEE Transaction on Multimedia on the 25th of December 2021, and is under peer review at the time of writing (July 2022). In this article, Exquisitor is extended with a priority queue to support incremental retrieval, and uses query optimisation policies based on information of filters and observed items within clusters in the eCP index. Note that this is an expansion and revision of the article presented in Section 3.1. To truly check Exquisitor's ability to find relevant items when filters are applied, the evaluation protocols from the article in

Section 3.2 are used. Since Exquisitor has a major focus on being efficient at scale, the relatively small collections from the article in Section 3.2 are combined with the large-scale YFCC100M collection. From the evaluation, we observe that without incremental retrieval and query optimisation policies (the initial Exquisitor version) the time per interaction round is the fastest. However, as the initial version only has a fixed scope, relevant items fall out of the scope when filters are applied which leads to no suggestions being returned, halting the interactive session. Enabling incremental retrieval, as expected expands the scope and manages to find relevant items, but it does increase the average latency to around 1 second. The increase in latency is alleviated through the various query optimisation policies. With these additions to Exquisitor, it is shown to be capable of supporting both exploration- and search-oriented tasks at scale in interactive times.

Majority of the work has primarily focused on the interactive learning and index performance with visual semantic concepts from images. For the evaluation with YFCC100M, the metadata from the images has been used to define LDA-topics as a representation for the textual modality. In Exquisitor each modality representation of visual semantic concepts and the LDA-Topics are stored in their own index. During the retrieval process, late fusion by rank aggregation is used to combine the results from each modality. There is a major disparity between the quality of the two modality representations, with some images not containing any viable metadata, and others having very little of it. With this disparity and a fusion approach that attempts to treat modalities equally, the suggestions from the textual modality have a high chance of being unrelated items. In such cases, it is safer to discard the representation from the interactive learning process and primarily use it for search or for filters. While semantic concepts can be a good representation for the visual content of an image, however, it may not be enough for a video. In a video many things can be occurring, which can relate to concepts, actions, scenery, or audio, all of which may be useful during the interactive learning process. Typically, in these cases, fusion is used to combine the information from the representations to get suggestions. However, the relation between modalities both in terms of quality and in terms of relevance to a given task can be difficult to determine.

Section 3.4 presents the article entitled **Influence of Late Fusion of High-Level Features on User Relevance Feedback for Videos**, which has been submitted to the 2nd International Workshop on Interactive Multimedia Retrieval (IMuR 2022) on the 7th of July 2022, and is under peer-review at the time of writing (July 2022). In this article, several late fusion methods with user relevance feedback have been used to solve tasks on three video collections, using the modalities of semantic concepts, actions, scenes, and audio. The first late fusion method uses a rank aggregation, where the result set from each modality representation is combined based on an aggregate score derived from their ranks in each representation. A variation of this is weighted rank aggregation, which depicts the case of a task leaning towards a specific modality and the user setting a preference for that modality, leading to its rank weighing more in the fusion. Other approaches considered are no fusion by dividing the suggestion set into top items from each modality, and partial fusion where the suggestion set consists of fused and non-fused items. The outcome from this work shows that fusion is beneficial, but the presence of weaker modalities can negatively effect the quality. Similarly, setting a preference on a modality is only good when it is on the right modality for a task. Partial fusion is generally better when it is difficult to determine a modality preference for a task. Based on this work, it is safe to say that including all modalities is not always good, and that modalities that work well for one collection, may not work well on others. Thus, when employing multiple modalities with Exquisitor, using partial fusion, along with giving the user an option for preferring modality, may improve the overall interactive learning experience.

Interactive Learning for Multimedia at Large

Omar Shahbaz Khan[1], Björn Þór Jónsson[1,4], Stevan Rudinac[2],
Jan Zahálka[3], Hanna Ragnarsdóttir[4], Þórhildur Þorleiksdóttir[4],
Gylfi Þór Guðmundsson[4], Laurent Amsaleg[5], and Marcel Worring[2]

[1] IT University of Copenhagen, Copenhagen, Denmark
[2] University of Amsterdam, Amsterdam, Netherlands
[3] Czech Technical University in Prague, Prague, Czech Republic
[4] Reykjavik University, Reykjavík, Iceland
[5] CNRS–IRISA, Rennes, France

Abstract. Interactive learning has been suggested as a key method for addressing analytic multimedia tasks arising in several domains. Until recently, however, methods to maintain interactive performance at the scale of today's media collections have not been addressed. We propose an interactive learning approach that builds on and extends the state of the art in user relevance feedback systems and high-dimensional indexing for multimedia. We report on a detailed experimental study using the ImageNet and YFCC100M collections, containing 14 million and 100 million images respectively. The proposed approach outperforms the relevant state-of-the-art approaches in terms of interactive performance, while improving suggestion relevance in some cases. In particular, even on YFCC100M, our approach requires less than 0.3 seconds per interaction round to generate suggestions, using a single computing core and less than 7GB of main memory.

Keywords: Large multimedia collections · Interactive multimodal learning · High-dimensional indexing · ImageNet · YFCC100M

1 Introduction

A dominant trend in multimedia applications for industry and society today is the ever-growing scale of media collections. As the general public has been given tools for unprecedented media production, storage and sharing, media generation and consumption have increased drastically in recent years. Furthermore, upcoming multimedia applications in countless domains—from smart urban spaces and business intelligence to health and wellness, lifelogging, and entertainment—increasingly require joint modelling of multiple modalities [20, 47]. Finally, users expect to be able to work very efficiently with large-scale collections, even with the limited computing resources they have at their immediate disposal. All these trends contribute to making scalability a greater concern than ever before.

User relevance feedback, a form of interactive learning, provides an effective mechanism for addressing various analytic tasks that require alternating between search and exploration. Figure 1 shows an example of such a relevance feedback

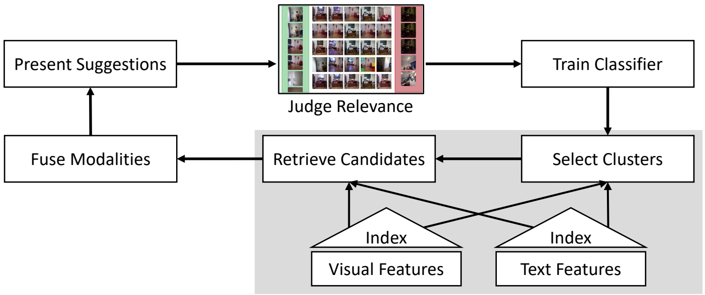


Fig. 1: An outline of the user relevance feedback approach proposed in this paper. The shaded area indicates that the traditional relevance feedback pipeline is enhanced with a novel query mechanism to a state-of-the-art cluster-based high-dimensional index.

process, where positive and negative relevance judgments from the user are used to train a classifier, which in turn is used to provide new suggestions to the user, with the process continuing until the user completes the interaction. There has been relatively little work on user relevance feedback and truly scalable and interactive multimedia systems in general in the last decade, however, which recently raised serious concerns in the multimedia community [39]. Clearly, the time has come to re-visit interactive learning with an aim towards scalability.

We propose Exquisitor, a highly scalable and interactive approach for user relevance feedback on large media collections. As illustrated in Figure 1, the proposed approach tightly integrates high-dimensional indexing with the interactive learning process. To the best of our knowledge, our approach is the first scalable interactive learning method to go beyond utilizing clustering in the preprocessing phase only. To evaluate the approach, we propose a new zero-shot inspired evaluation protocol over the ImageNet collection, and use an existing protocol for the large-scale YFCC100M collection. We show that our approach outperforms state-of-the-art approaches in terms of both suggestion relevance and interactive performance. In particular, our approach requires less than 0.3 seconds per interaction round to generate suggestions from the YFCC100M collection, using a single CPU core and less than 7GB of main memory.

The remainder of this paper is organized as follows. In Section 2, we discuss interactive learning from a scalability perspective, setting the stage for the novel approach. In Section 3, we then present the proposed approach in detail, and compare its performance to the state of the art in Section 4, before concluding.

2 Related Work

As outlined in the introduction, combining interactive learning with high dimensional indexing is a step towards unlocking the true potential of multimedia collections and providing added value for users. In this section we first describe the state of the art in interactive learning. Then, based on the identified advantages and limitations of interactive learning algorithms, we provide a set of requirements that high-dimensional indexing should satisfy for facilitating interactivity on extremely large collections. Finally, we use those requirements for reflecting on the state of the art in high-dimensional indexing.

Interactive Learning: Interactive learning has long been a cornerstone of facilitating access to document collections [1, 18, 27, 16] and it became an essential tool of multimedia researchers from the early days of content-based image and video retrieval [36, 15]. The most popular flavour of interactive learning is user relevance feedback that presents the user, in each interaction round, with the items for which the classification model is most confident [36]. User relevance feedback has frequently been used in the best performing entries of benchmarks focusing on interactive video search and exploration [28, 41]. However, those solutions were designed for collections far smaller than YFCC100M, which is the challenge we take in this paper. Linear models for classification, such as Linear SVM are still amongst the most frequent choices in relevance feedback applications [22, 31, 48] due to their simplicity, interpretability and explainability as well as the ability to produce accurate results with few annotated samples and scale to very large collections.

To the best of our knowledge, Blackthorn [48] is the most efficient interactive multimodal learning approach in the literature. Its efficiency is achieved through adaptive data compression and feature selection, multi-core processing, and a classification model capable of scoring items directly in the compressed domain. Compared to product quantization [17], a popular alternative optimized for k-NN search, Blackthorn was found to yield significantly more accurate results over YFCC100M with similar latency (1.2 seconds), while consuming modest computational resources (16 CPU cores with 5 GB of main memory).

Indexing Requirements: We have identified the following requirements for high-dimensional indexing to enhance the performance of interactive learning:

R1 *Short and Stable Response Time:* A successful indexing approach in interactive learning combines good result quality with response time guarantees [44].

R2 *Preservation of Feature Space Similarity Structure:* Since interactive classifiers compute relevance based on a similarity structure on the feature space, the space partitioning of the high-dimensional indexing algorithm must preserve this similarity structure.

R3 *k Farthest Neighbours:* Relevance feedback approaches typically try to inform the user by presenting the most confidently relevant items based on the judgments observed so far, which are the items farthest from the classification boundary. As results are intended for display on screen, the index should thus return k farthest neighbours (k-FN).

We are not aware of any work in the high-dimensional literature targeting approximate k-FN where the query is a classification boundary. We therefore next review the related work and discuss how well different classes of high-dimensional indexing methods can potentially satisfy these three requirements.

High-Dimensional Indexing Scalable high-dimensional indexing methods generally rely on approximation through some form of quantization. One class of methods uses scalar quantization. The NV-tree, for example, is a large-scale index that uses random projections at its core [25, 26], recursively projecting points onto segmented random lines. LSH is another indexing method that uses random projections acting as locality preserving hashing functions [2, 8]. Recently, multimedia researchers have considered hashing for multimedia applications, but typically at a much smaller scale than considered here [13, 29, 42]. LSH has been considered in the context of hyperplane-based nearest-neighbour queries [5, 45] and point-based farthest-neighbour queries [7, 32, 46], but not in the context of *hyperplane-based farthest-neighbour* queries. We argue that LSH and related methods fail to satisfy the three requirements above: they focus on quality guarantees rather than performance guarantees (**R1**); hashing creates "slices" in high-dimensional space, making ranking based on distance to a decision boundary difficult (**R2**); and they typically focus on ϵ-range queries, giving no guarantees on the number of results returned (**R3**).

A second class of methods is based on vector quantization, typically using clustering approaches, such as k-means, to determine a set of representative feature vectors to use for the quantization. These methods create Voronoï cells in the high-dimensional space, which satisfy **R2** well. Some methods, such as BoW-based methods, only store image identifiers in the clusters, thus failing to support **R3**, while others store the entire features, allowing to rank the results from the farthest clusters. Finally, many clustering methods seek to match well the distribution of data in the high-dimensional space. Typically, these methods end with a large portion of the collection in a single cluster, which in turn takes very long to read and score, thus failing to satisfy **R1** [12].

Product quantization (PQ) [17] and its variants [4, 10, 14] cluster the high-dimensional vectors into low-dimensional subspaces that are indexed independently. PQ better captures the location of points in the high-dimensional space, which in turn improves the quality of the approximate results that are returned. One of the main aims of PQ is data compression, however, and PQ-based methods essentially transform the Euclidean space, complicating the identification of furthest neighbours (**R2**). PQ-compression was compared directly with the Blackthorn compression method designed for interactive learning [48] and was shown as having inferior performance. The extended Cluster Pruning (eCP) algorithm [11, 12], however, is an example of a vector quantifier which attempts to balance cluster sizes for improved performance, thus aiming to satisfy all three requirements; we conclude that eCP is our prime candidate.

3 The Exquisitor Approach

In this section, we describe Exquisitor, a novel interactive learning approach that tightly integrates high-dimensional indexing with the interactive learning process, facilitating interactive learning at the scale of the YFCC100M image collection using very moderate hardware resources. Figure 1 shows an outline of the Exquisitor approach. We start by considering the multimodal data representation and classifier, before describing the indexing and retrieval algorithms in separate sub-sections. To facilitate the exposition in this section, we occasionally use actual examples from the YFCC100M collection.

3.1 Media Representation and Classification Model

Similar to [48], we choose to represent each image with two semantic feature vectors, one for visual content using deep-learning-based feature vectors and the second for textual content by extracting LDA topics from any textual metadata associated with the images. Although more descriptive approaches for extracting text features exist, in this case the LDA is effective in yielding discriminative representation for different items.

Directly working with these representations, however, is infeasible. In our case, using 1,000 and 100 dimensions for the visual and text domains, respectively, the feature vectors would require 8.8KB of main memory per image, or around 880GB for the YFCC100M collection, which is far beyond the storage capacity of typical hardware. We use the data compression method presented in [48] that preserves semantic information with over 99% compression rate.

Consistent with the state of the art in user relevance feedback, the classifier used in Exquisitor is Linear SVM. The choice is further motivated by the algorithm's speed, reasonable performance and compatibility with the sparse compressed representation. Note that the choice of interactive classifier and features in each respective modality made in this paper is not an inherent setting of Exquisitor; they can be replaced as deemed fit. The choices made in this paper are in line with the choices made in the state of the art Exquisitor competes against (most notably [48]), providing a level field for experimental evaluation.

3.2 Data Indexing

The data indexing algorithm used in Exquisitor is based on the extended Cluster Pruning (eCP) algorithm [12]. As motivated in Section 2, the goal is to individually cluster each of the two feature representations with a vector quantizer, using a hierarchical index structure to facilitate efficient selection of clusters to process for suggestions. For each collection, cluster representatives are selected randomly and clusters are formed by assigning images to the nearest cluster based on Euclidean distance, computed efficiently directly in compressed space. The indexing algorithm recursively selects 1% of the images at each level as representatives for the level above, until fewer than 100 representatives remain to form the root of the index. As an example, the bottom level of the index for each modality in the YFCC100M collection consists of $992,066$ clusters, organized in a 3 level deep index hierarchy, which gives on average 100 images per cluster and per internal node. When building the indices, the average cluster size was chosen to be small, as previous studies show that searching more small clusters yields better results than searching fewer large clusters [11, 40].

3.3 Suggestion Retrieval

The retrieval of suggestions has the following three phases: identify b most relevant clusters, select r most relevant candidates per modality, and fuse modalities to retrieve k most relevant suggestions.[6]

Identify b Most Relevant Clusters: In each interaction round, the index of representatives is used to identify, for each modality, the b clusters most likely to contain useful candidates for suggestions. This search expansion parameter, b, affects the size of the subset that will be scored and can be used to balance between search quality and latency at run-time. All cluster representatives are scored by the interactive classifier and the b clusters farthest from the separating plane in the positive direction are selected as the most relevant clusters. In Section 4.3 we evaluate the effects of b on the YFCC100M collection.

We observe that with the YFCC100M collection, both modalities have 1-2 clusters that are very large, with more than 1M items. These clusters require a significant effort to process, without improving suggestion quality. In the experiments reported here, we have therefore omitted clusters larger than 1M.

Select r Most Relevant Candidates per Modality: Once the most relevant b clusters have been identified, the compressed feature vectors within these clusters are scored to suggest the r most relevant media items for each modality. The method of scoring individual feature vectors is the same as when selecting the most relevant clusters.

Some notes are in order here. First, in this scoring phase, media items seen in previous rounds are not considered candidates for suggestions. Second, an item already seen in the first modality is not considered as a suggestion in the second modality. Third, if all b clusters are small, the system may not be able to identify r candidates, in which case it simply returns all the candidates found. Finally, we observe that treating all b clusters equally results in an over-emphasis on items that score very highly in only one modality, but have a low score in the other modality. This can be troublesome if the relevant items have a decent score in both modalities. By segmenting the b clusters into S_c segments of size b/S_c this dominance can be avoided; we explore the impact of S_c in Section 4.3.

Modality Fusion for k Most Relevant Suggestions: Once the r most relevant candidates from each modality have been identified, the modalities must be fused by aggregating the candidate lists to produce the final list of k suggestions. First, for each candidate in one modality, the score in the other modality is computed if necessary, by directly accessing the compressed feature vector, resulting in $2r$ candidates with scores in both modalities.[7] Second, the rank of each item in each modality is computed by sorting the $2r$ candidates. Finally, the average rank is used to produce the final list of suggestions.

[6] In the case of unimodal retrieval, the latter two phases can be merged.

[7] To facilitate late modality fusion, the location of each feature vector in each cluster index is stored; each vector requires ~800KB of RAM for the YFCC100M collection.

Multi-Core Processing: If desired, Exquisitor can take advantage of multiple CPU cores. With w cores available, the system creates w worker processes and assigns b/w clusters to each worker. Each worker produces r suggestions in each modality and fuses the two modalities into k candidates, as described above. The top k candidates overall are then selected by repeating the modality fusion process for the suggestions produced by the workers.

4 Experimental Evaluation

In this section, we experimentally analyse the interactive performance of Exquisitor. We first outline the baseline comparison architectures from the literature. We then describe two detailed experiments. In the first experiment, we propose a new experimental protocol for interactive learning based on the popular ImageNet benchmark dataset, and show that a) the Linear SVM model is capable of discovering new classes in the data, and b) with high-dimensional indexing, performance is significantly improved. In the second experiment, we then use a benchmark experimental protocol from the literature defined over the YFCC100M collection, and show that at this scale the Exquisitor approach outperforms the baseline architectures significantly, both in terms of retrieval quality and interactive performance.

4.1 Baseline Approaches

In the experiments we compare Exquisitor with the following state-of-the-art approaches from the literature.

Blackthorn: To the best of our knowledge, Blackthorn [48] is the only direct competitor in the literature for interactive learning at the YFCC100M scale. Unlike Exquisitor, Blackthorn uses no indexing or prior knowledge about the structure of the collection, instead using data compression and multi-core processing for scalability.

kNN+eCP: This baseline is representative of pure query-based approaches using a k-NN query vector based on relevance weights [34, 23], an approach that was initially introduced for text retrieval [35] but has been adapted for CBIR with relevance feedback [37].

SVM+LSH, kNN+LSH: These baselines represent SVM-based and k-NN-based approaches using LSH indexing. We replace the eCP index with a multi-probing LSH index [30] using the FALCONN library [3].

All comparison architectures are compiled with g++. Experiments are performed using dual 8-core 2.4 GHz CPUs, with 64GB RAM and 4TB local SSD storage. Note, however, that even the YFCC100M collection requires less than 7GB of SSD storage and RAM, and most experiments use only a single CPU core.

While tuning LSH performance is difficult, due to the many parameters that interact in complex ways (L is the number of tables, B is the number of buckets in each table, and p is the number of buckets to read from each table at query time), we have strived to find parameter settings that a) lead to a similar cell size distribution as eCP and b) yield the best performance.

4.2 Experiment 1: Discovering ImageNet Concepts

Zero-shot learning is a method which trains a classifier to find target classes without including the target classes when training the model. Taking inspiration from zero-shot learning, the objective of this experiment is to simulate a user that is looking for a concept that is on their mind, but is not directly represented in the data; a successful interactive learning approach should be able to do this.

Image Collection: ImageNet is an image database based on the WordNet hierarchy. It is a well-curated collection targeting object recognition research as the images in the collection are categorized into approximately 21,000 WordNet synsets (synonym sets) [9]. The collection contains 14,198,361 images, each of which is represented with the 1,000 ILSVRC concepts [38]. Due to images being categorized into multiple WordNet synsets, the ImageNet collection contains duplicate images, each labelled differently, which can lead to false negatives.

Experimental Protocol: The protocol for the experiment is constructed by randomly selecting 50 concepts from the 1,000 ILSVRC concepts. For each concept a simulated user (henceforth called actor) is created, which knows which images belong to its concept and is charged with the task of finding items belonging to that concept. We have then created and indexed 5 different collections of visual features, where the feature value of the concepts belonging to 10 different actors have been set to 0 to introduce the zero-shot setting.

The workload for each actor proceeds as follows. Initially, 10 images from the concept and 100 random images are used as positive and negative examples, respectively, to create the first round of suggestions, simulating a situation where the exploration process has already started. In each round of the interactive learning process, the actor considers the suggested images from the system and designates images from its concept as positive examples, while 100 additional negative examples are drawn randomly from the entire collection. This is repeated for 10 interaction rounds, with performance statistics collected in each round. To combat the duplicate images problem, we first run the workload using the original data where the concepts are known in order to establish an upper bound baseline for each approach.

Results: Figure 2 compares the average precision across the 10 rounds for each of the approaches under study, for both the case when the concept is *known* (blue columns) and *unknown* (red columns). For Exquisitor and eCP+kNN, the search expansion parameter b is set to 256, while SVM+LSH and kNN+LSH have the following settings for the LSH index: $L = 10$, $B = 2^{14}$, and $p = 20$.

Overall, the figure shows that precision for the known case is nearly 50% on average for the SVM-based approaches, and only slightly lower for the k-NN-based approaches. When the feature value for the actor's concept is not known, however, the average precision drops only slightly for the SVM-based approaches, while the k-NN-based approaches perform very poorly. These results indicate that the Linear-SVM model is clearly superior to the k-NN approach.

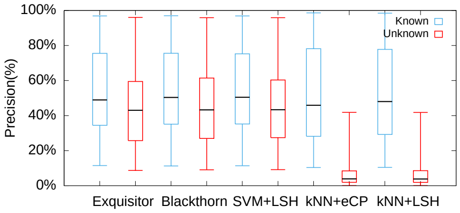


Fig. 2: Average precision per round across all ImageNet actors for each interactive learning approach. The blue columns depict the known case, while the red depict the unknown case.

Table 1: Average latency per interaction round across all ImageNet actors.

Approach	Latency
Exquisitor	0.008 s
Blackthorn (1w)	0.130 s
Blackthorn (16w)	0.017 s
SVM+LSH	0.008 s
kNN+eCP	0.008 s
kNN+LSH	0.004 s

Turning to the average time required for each iteration of the learning process, Table 1 compares the approaches under study. Overall, we note that the four approaches relying on high-dimensional indexing perform very well using a single computing core, requiring less than 10 milliseconds to return suggestions. At the moderate scale of the ImageNet collection, eCP and LSH perform similarly. Running Blackthorn with 16 cores is 2x slower, however, while running Blackthorn using a single core is about 16x slower.

As mentioned above, precision is impacted by the ImageNet collection itself containing duplicates. A visual inspection of the results of some of the worst-performing actors suggest that with known data, the majority of the non-relevant images are such duplicates. For the unknown case, a similar trend is seen for the SVM-based approaches, but not for the k-NN-based approaches, which clearly are unable to steer the query vectors for suggestions to a more relevant part of the collection. Figure 3 shows some examples of this, for the actor for concept “knee pad”. As the figure shows, with any SVM-based approach the irrelevant images are also knee pads, but tagged to another related concept, while for the k-NN-based approach, no relevant images were found and the irrelevant images bear no relationship to knee pads.

4.3 Experiment 2: Performance at YFCC100M Scale

The goal of this experiment is to study the scalability of the Exquisitor approach, in comparison to the baseline approaches from the literature. To that end, we apply the only interactivee learning evaluation protocol from the literature that we are aware of at YFCC100M scale [48].

Collection: The YFCC100M collection contains 99,206,564 Flickr images, their associated annotations (i.e. title, tags and description), a range of metadata produced by the capturing device, the online platform, and the user (e.g., geolocation and time stamps). The visual content is represented using the 1,000 ILSVRC concepts [38] extracted using the GoogLeNet convolutional neural network [43]. The textual content is encoded by a) treating the title, tags, and

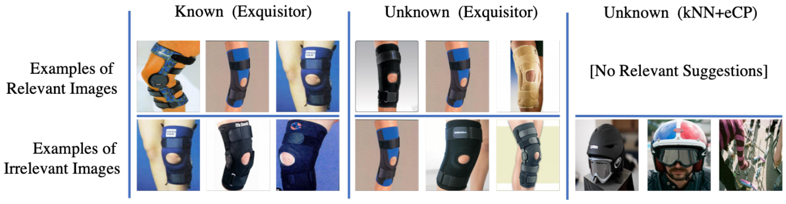

Known (Exquisitor)
Unknown (Exquisitor)
Unknown (kNN+eCP)
Examples of Relevant Images
Examples of Irrelevant Images
[No Relevant Suggestions]

Fig. 3: Examples of relevant and irrelevant suggestions for different approaches for the ImageNet actor for the concept “knee pad”.

description as a single text document, and b) extracting 100 LDA topics for each image using the gensim toolkit [33].

The YFCC100M collection, being large and uncurated, displays some interesting phenomena worth mentioning. First, a non-trivial proportion of images are a standard Flickr “not found” image.[8] A similar situation arises in the text modality, with many images lacking text information altogether, resulting in zero-valued vectors. Such images are essentially noise, potentially crowding out more suitable candidates. Second, with the collection being massive and the data being compressed and clustered, discriminativeness of feature vectors becomes a problem: non-identical images may be mapped to identical feature vectors.

Experimental Protocol: For this experiment we follow the experimental interactive learning protocol in [48]. This evaluation protocol is inspired by the MediaEval Placing Task [24, 6], in which actors simulating user behaviour look for images from 50 world cities.

To illustrate the tradeoffs between the interactive performance and result quality, we focus our analysis on precision and latency (response time) per interaction round. It is worth noting that due to both the scale of YFCC100M and its unstructured nature, precision is lower than in experiments involving small and well-curated collections.

Impact of Search Expansion Parameter: We start by exploring the impact of the search expansion parameter b for the eCP index. Figure 4 analyses the impact of b, the number of clusters read and scored, on the precision (fraction of relevant items seen) in each round of the interactive exploration. The x-axis shows how many clusters are read for scoring at each round, ranging from $b = 1$ to $b = 512$ (note the logarithmic scale of the axis), while the y-axis shows the average precision across the first 10 rounds of analysis. The figure shows precision for two Exquisitor variants, with $S_c = 1$ and $S_c = 16$. In both cases, only one worker is used, $w = 1$. For comparison, the figure also shows the average precision for Blackthorn, the state-of-the-art SVM-based alternative.

As Figure 4 shows, result quality is surprisingly good when scoring only a single cluster in each interaction round, returning about two-thirds of the pre-

[8] The image collection was actually downloaded very shortly after release, but already then this had become a significant issue.

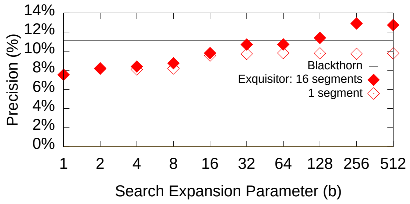


Fig. 4: Average precision over 10 rounds of analysis across all YFCC100M actors. Exquisitor: Varying b; $w = 1$; $S_c = 1, 16$.

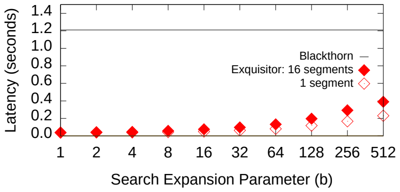


Fig. 5: Average latency over 10 rounds of analysis across all YFCC100M actors. Exquisitor: Varying b; $w = 1$; $S_c = 1, 16$.

cision of the state-of-the-art algorithm. As more clusters are considered, quality then improves further. As expected, dividing the b clusters into $S_c = 16$ chunks results in better quality, an effect that becomes more pronounced as b grows. In particular, with $b = 256$, Exquisitor returns significantly better results than Blackthorn. The reason is that by assigning the b relevant clusters to $S_c = 16$ segments, Exquisitor is able to emphasize the bi-modal media items as explained in Section 3.3. Note that as further clusters are added with Exquisitor ($b = 512$ and beyond), the results become more and more similar to the Blackthorn results.

Figure 5, on the other hand, shows the latency per interaction round. The figure again shows the two Exquisitor variants, with $S_c = 1$ and $S_c = 16$; in both cases, one worker is used, $w = 1$. For comparison, as before, it also shows the average latency for Blackthorn (with 16 CPU cores). Unsurprisingly, Figure 5 shows linear growth in latency with respect to b (recall the logarithmic x-axis). With $b = 256$, each interaction round takes less than 0.3 seconds with $S_c = 16$, and about 0.17 seconds with $S_c = 1$. Both clearly allow for interactive performance; the remainder of our experiments focus on $b = 256$. If even shorter latency is desired, however, fewer clusters can be read: $b = 32$, for example, also gives a good tradeoff between latency and result quality. This latency is produced using only a single CPU core, meaning that the latency is $\sim$4x better than Blackthorn, with 16x fewer computing cores, for an improvement of $\sim$64x, or nearly two orders of magnitude. With this knowledge we see b as a parameter that is determined by collection size and the task a user is dealing with, but, as a general starting point we recommend $b = 256$ for large collections.

Comparison: Figure 6 shows the tradeoff between result quality, measured by average precision across 10 rounds of interaction, and the average latency required to produce the suggestions in each round. For Exquisitor, the figure essentially summarizes Figures 4 and 5. For kNN+eCP, the dots represent the same b parameter values, while for the LSH-based approaches a variety of parameter values are represented. The figure clearly demonstrates that Exquisitor is the best approach in both precision and response time compared to all the baseline approaches, achieving better precision than Blackthorn, requiring less than 0.3 seconds compared to Blackthorn's 1.2 seconds. Both k-NN-bases approaches get

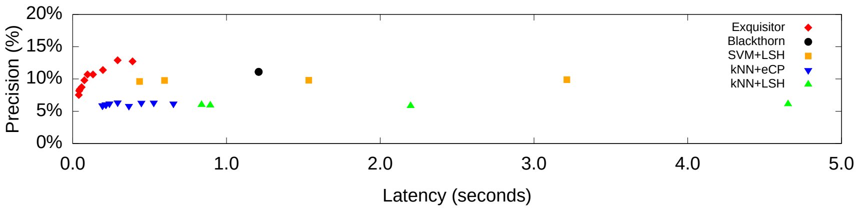


Fig. 6: Average precision vs. latency over 10 rounds of analysis across all YFCC100M actors. Exquisitor, kNN+eCP: $b = 1 - 512$. LSH: $L = 10$, $B = [2^{10}, 2^{18}]$, $p = [15, 40]$.

stuck at 6% which is to be expected since the k-NN query narrows down the scope of the search making it impossible to get out of local optima. SVM+LSH performs better, with precision nearly as good as Blackthorn and response time close to Exquisitor. Overall, however, Exquisitor performs better partly due to being able to utilize the SVM during cluster selection with k-FN queries, and partly due to the cluster segments allowing better multi-modal results.

5 Conclusions

In this paper, we presented Exquisitor, a new approach for exploratory analysis of very large image collections with modest computational requirements. Exquisitor combines state-of-the-art large-scale interactive learning with a new cluster-based retrieval mechanism, enhancing the relevance capabilities of interactive learning by exploiting the inherent structure of the data. Through experiments conducted on YFCC100M, the largest publicly available multimedia collection, Exquisitor achieves higher precision and lower latency, with less computational resources. Additionally, through a modified zero-shot learning experiment on ImageNet, we determine the Exquisitor approach to be excellent at solving cumbersome classification tasks. Exquisitor also introduces customizability that is, to the best of our knowledge, previously unseen in large-scale interactive learning by: (i) allowing a tradeoff between low latency (few clusters) and high quality (many clusters); and (ii) combatting data skew by omitting huge (and thus likely nondescript) clusters from consideration. Exquisitor has recently been used successfully in interactive media retrieval competitions such as the Lifelog Search Challenge [21] and Video Browser Showdown [19]. In conclusion, Exquisitor provides excellent performance on very large collections while being efficient enough to bring large-scale multimedia analytics to standard desktops and laptops, and even high-end mobile devices.

Acknowledgments: This work was supported by a PhD grant from the IT University of Copenhagen and by the European Regional Development Fund (project Robotics for Industry 4.0, CZ.02.1.01/0.0/0.0/15 003/0000470).

References

1. Allan, J.: Incremental relevance feedback for information filtering. In: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 270–278. ACM, New York, NY, USA (1996)
2. Andoni, A., Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In: Proceedings of the IEEE Symposium on the Foundations of Computer Science. pp. 459–468. IEEE Computer Society, Berkeley, CA, USA (2006)
3. Andoni, A., Indyk, P., Laarhoven, T., Razenshteyn, I., Schmidt, L.: Practical and optimal lsh for angular distance. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28, pp. 1225–1233. Curran Associates, Inc. (2015)
4. Babenko, A., Lempitsky, V.S.: The inverted multi-index. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(6), 1247–1260 (2015)
5. Basri, R., Hassner, T., Zelnik-Manor, L.: Approximate nearest subspace search. IEEE Trans. Pattern Anal. Mach. Intell. **33**(2), 266–278 (2011)
6. Choi, J., Hauff, C., Laere, O.V., Thomee, B.: The placing task at mediaeval 2015. In: Proceedings of the MediaEval 2015 Workshop. CEUR, Wurzen, Germany (2015)
7. Curtin, R.R., Gardner, A.B.: Fast approximate furthest neighbors with data-dependent candidate selection. In: Proc. SISAP. pp. 221–235. Springer, Tokyo, Japan (2016)
8. Datar, M., Immorlica, N., Indyk, P., Mirrokni, V.S.: Locality-sensitive hashing scheme based on p-stable distributions. In: Proc. ACM Symposium on Computational Geometry. pp. 253–262. ACM, Brooklyn, NY, USA (2004)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255 (2009)
10. Ge, T., He, K., Ke, Q., Sun, J.: Optimized product quantization. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(4), 744–755 (2014)
11. Gudmundsson, G.Þ., Amsaleg, L., Jónsson, B.Þ.: Impact of storage technology on the efficiency of cluster-based high-dimensional index creation. In: Proc. International Conference on Database Systems for Advanced Applications (DASFAA). pp. 53–64. Springer, Busan, South Korea (2012)
12. Gudmundsson, G.Þ., Jónsson, B.Þ., Amsaleg, L.: A large-scale performance study of cluster-based high-dimensional indexing. In: Proc. International Workshop on Very-large-scale Multimedia Corpus, Mining and Retrieval (VLS-MCMR). pp. 31–36. ACM, Firenze, Italy (2010)
13. Hansen, C., Hansen, C., Simonsen, J.G., Alstrup, S., Lioma, C.: Unsupervised neural generative semantic hashing. In: Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 735–744. SIGIR'19, ACM, New York, NY, USA (2019)
14. Heo, J., Lin, Z., Yoon, S.: Distance encoded product quantization. In: Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition. pp. 2139–2146. IEEE Computer Society, Columbus, OH, USA (2014)
15. Huang, T., Dagli, C., Rajaram, S., Chang, E., Mandel, M., Poliner, G.E., Ellis, D.: Active learning for interactive multimedia retrieval. Proc. IEEE **96**(4), 648–667 (2008)

16. Iwayama, M.: Relevance feedback with a small number of relevance judgements: Incremental relevance feedback vs. document clustering. In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 10–16. ACM, New York, NY, USA (2000)
17. Jégou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(1), 117–128 (2011)
18. Joachims, T.: A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In: Proceedings of the Fourteenth International Conference on Machine Learning. pp. 143–151. ICML '97, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1997)
19. Jónsson, B.Þ., Khan, O.S., Koelma, D.C., Rudinac, S., Worring, M., Zahálka, J.: Exquisitor at the video browser showdown 2020. In: International Conference on Multimedia Modeling. pp. 796–802. Springer (2020)
20. Jónsson, B.Þ., Worring, M., Zahálka, J., Rudinac, S., Amsaleg, L.: Ten research questions for scalable multimedia analytics. In: International Conference on Multimedia Modeling. pp. 290–302. Springer (2016)
21. Khan, O.S., Jónsson, B.Þ., Zahálka, J., Rudinac, S., Worring, M.: Exquisitor at the lifelog search challenge 2019. In: Proceedings of the ACM Workshop on Lifelog Search Challenge. pp. 7–11. ACM (2019)
22. Kovashka, A., Parikh, D., Grauman, K.: Whittlesearch: Interactive image search with relative attribute feedback. International Journal of Computer Vision **115**(2), 185–210 (2015)
23. Lan, M., Tan, C.L., Su, J., Lu, Y.: Supervised and traditional term weighting methods for automatic text categorization. IEEE transactions on pattern analysis and machine intelligence **31**(4), 721–735 (2008)
24. Larson, M., Soleymani, M., Serdyukov, P., Rudinac, S., Wartena, C., Murdock, V., Friedland, G., Ordelman, R., Jones, G.J.F.: Automatic tagging and geotagging in video collections and communities. In: Proceedings of the 1st ACM International Conference on Multimedia Retrieval. pp. 51:1–51:8. ACM, New York, NY, USA (2011)
25. Lejsek, H., Ásmunðsson, F.H., Jónsson, B.Þ., Amsaleg, L.: NV-Tree: An efficient disk-based index for approximate search in very large high-dimensional collections. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(5), 869–883 (2009)
26. Lejsek, H., Jónsson, B.Þ., Amsaleg, L.: NV-Tree: nearest neighbors at the billion scale. In: Proceedings of the ACM International Conference on Multimedia Retrieval. ACM, Trento, Italy (2011)
27. Lewis, D.D., Schapire, R.E., Callan, J.P., Papka, R.: Training algorithms for linear text classifiers. In: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 298–306. SIGIR '96, ACM, New York, NY, USA (1996)
28. Lokoč, J., Bailer, W., Schoeffmann, K., Muenzer, B., Awad, G.: On influential trends in interactive video retrieval: Video browser showdown 2015–2017. IEEE Transactions on Multimedia **20**(12), 3361–3376 (2018)
29. Lu, X., Zhu, L., Cheng, Z., Nie, L., Zhang, H.: Online multi-modal hashing with dynamic query-adaption. In: Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 715–724. ACM, New York, NY, USA (2019)

30. Lv, Q., Josephson, W., Wang, Z., Charikar, M., Li, K.: Multi-probe lsh: efficient indexing for high-dimensional similarity search. In: Proceedings of the 33rd international conference on Very large data bases. pp. 950–961. VLDB Endowment (2007)
31. Mironică, I., Ionescu, B., Uijlings, J., Sebe, N.: Fisher kernel temporal variation-based relevance feedback for video retrieval. Computer Vision and Image Understanding **143**, 38–51 (2016)
32. Pagh, R., Silvestri, F., Sivertsen, J., Skala, M.: Approximate furthest neighbor with application to annulus query. Inf. Syst. **64**, 152–162 (2017)
33. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010)
34. Robertson, S.E., Spärck Jones, K.: Simple, proven approaches to text retrieval. Tech. rep., University of Cambridge, Computer Laboratory (1994)
35. Rocchio, J.J.: Relevance feedback in information retrieval. Tech. rep., University of Harvard, Computer Laboratory (1965)
36. Rui, Y., Huang, T.S., Mehrotra, S.: Content-based image retrieval with relevance feedback in MARS. In: Proc. International Conference on Image Processing (ICIP). pp. 815–818. IEEE Computer Society, Santa Barbara, CA, USA (1997)
37. Rui, Y., Huang, T.S., Mehrotra, S.: Content-based image retrieval with relevance feedback in mars. In: Proceedings of International Conference on Image Processing. vol. 2, pp. 815–818. IEEE (1997)
38. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3), 211–252 (Dec 2015)
39. Schoeffmann, K., Bailer, W., Gurrin, C., Awad, G., Lokoč, J.: Interactive video search: Where is the user in the age of deep learning? In: Proc ACM Multimedia. pp. 2101–2103. ACM, Seoul, Republic of Korea (2018)
40. Sigurðardóttir, R., Hauksson, H., Jónsson, B.Þ., Amsaleg, L.: The quality vs. time tradeoff for approximate image descriptor search. In: Proc. IEEE EMMA workshop. IEEE, Tokyo, Japan (2005)
41. Snoek, C., Worring, M., de Rooij, O., van de Sande, K., Yan, R., Hauptmann, A.: Videolympics: Real-time evaluation of multimedia retrieval systems. IEEE MM **15**(1), 86–91 (2008)
42. Sun, C., Song, X., Feng, F., Zhao, W.X., Zhang, H., Nie, L.: Supervised hierarchical cross-modal hashing. In: Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 725–734. ACM, New York, NY, USA (2019)
43. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proc. IEEE CVPR. pp. 1–9. IEEE Computer Society, Boston, MA, USA (2015)
44. Tavenard, R., Jégou, H., Amsaleg, L.: Balancing clusters to reduce response time variability in large scale image search. In: International Workshop on Content-Based Multimedia Indexing. IEEE, Madrid, Spain (2011)
45. Vijayanarasimhan, S., Jain, P., Grauman, K.: Hashing hyperplane queries to near points with applications to large-scale active learning. IEEE Trans. Pattern Anal. Mach. Intell. **36**(2), 276–288 (2014)
46. Xu, X., Bao, J., Yao, B., Zhou, J., Tang, F., Guo, M., Xu, J.: Reverse furthest neighbors query in road networks. J. Comput. Sci. Technol. **32**(1), 155–167 (2017)

47. Zahálka, J., Worring, M.: Towards interactive, intelligent, and integrated multimedia analytics. In: Proc. of the IEEE Conference on Visual Analytics Science and Technology (VAST). pp. 3–12. Paris, France (2014)
48. Zahálka, J., Rudinac, S., Jónsson, B.T., Koelma, D.C., Worring, M.: Blackthorn: Large-scale interactive multimodal learning. IEEE Transactions on Multimedia **20**(3), 687–698 (2018)

Impact of Interaction Strategies on User Relevance Feedback

Omar Shahbaz Khan
IT University of Copenhagen
Copenhagen, Denmark
omsh@itu.dk

Björn Þór Jónsson
IT University of Copenhagen
Copenhagen, Denmark
bjth@itu.dk

Jan Zahálka
Czech Technical University in Prague
Prague, Czech Republic
jan.zahalka@cvut.cz

Stevan Rudinac
University of Amsterdam
Amsterdam, Netherlands
s.rudinac@uva.nl

Marcel Worring
University of Amsterdam
Amsterdam, Netherlands
m.worring@uva.nl

ABSTRACT

User Relevance Feedback (URF) is a class of interactive learning methods that rely on the interaction between a human user and a system to analyze a media collection. To improve URF system evaluation and design better systems, it is important to understand the impact that different interaction strategies can have. Based on the literature and observations from real user sessions from the Lifelog Search Challenge and Video Browser Showdown, we analyze interaction strategies related to (a) labeling positive and negative examples, and (b) applying filters based on users' domain knowledge. Experiments show that there is no single optimal labeling strategy, as the best strategy depends on both the collection and the task. In particular, our results refute the common assumption that providing more training examples is always beneficial: strategies with a smaller number of prototypical examples lead to better results in some cases. We further observe that while expert filtering is unsurprisingly beneficial, aggressive filtering, especially by novice users, can hinder the completion of tasks. Finally, we observe that combining URF with filters leads to better results than using filters alone.

CCS CONCEPTS

• **Information systems → Multimedia and multimodal retrieval**; **Evaluation of retrieval results**.

KEYWORDS

Interactive Learning, User Relevance Feedback, Multimedia Retrieval, Interaction Strategies

ACM Reference Format:

Omar Shahbaz Khan, Björn Þór Jónsson, Jan Zahálka, Stevan Rudinac, and Marcel Worring. 2021. Impact of Interaction Strategies on User Relevance Feedback. In *Proceedings of the 2021 International Conference on Multimedia Retrieval (ICMR '21), August 21–24, 2021, Taipei, Taiwan.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3460426.3463663

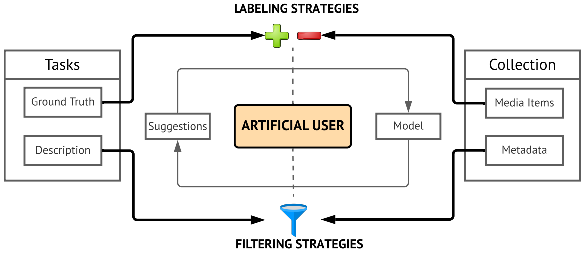


Figure 1: Our proposal for an automated evaluation process to understand the impact of user interaction strategies: We create artificial users that label suggestions from the collections using different strategies and apply metadata filters based on different levels of domain knowledge.

1 INTRODUCTION

Interactive Learning (IL) is a multimedia retrieval approach, where a user and system work together to build a model in order to satisfy a user information need [8, 9, 16, 19, 36, 37]. The system presents users with media items from a collection that they label as positive or negative, only positive [25], or only negative [31]. The labeled items are then used to train a classifier that is deployed to retrieve new suggestions. Some IL systems provide additional features, such as: filters to focus on subcollections; text search to find positive examples; or advanced browsing features, such as an event timeline. IL is a continuous process that stops when the user considers the task to be complete. User Relevance Feedback (URF) is a variation of IL that focuses on quick convergence of the user's information need by providing the most relevant items from the system's model in each interaction round. Active Learning, another variation of IL, suggests items from the collection that are most valuable to improve the model, rather than the most relevant ones [12]. Since we are focusing on scenarios where the user is seeking relevant items, we focus on URF.

Traditionally, evaluation of URF focuses on measuring the classification performance of the model used to retrieve items [11, 29, 34, 35]. Recently, automated evaluation protocols have been used to evaluate URF systems with a focus on artificial users [16, 23, 36] for large multimedia collections. However, these are based on *one* interaction strategy where the artificial user labels relevant items

ICMR '21, August 21–24, 2021, Taipei, Taiwan.
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8463-6/21/08…$15.00
https://doi.org/10.1145/3460426.3463663

as positives and the system labels everything else as negatives. Intuitively, when using URF systems, a common assumption is that more positive examples build a better model.

We conjecture that for many tasks it may be beneficial to have a smaller set of stronger positive examples. Understanding when to prefer one interaction strategy over another can greatly reduce the time of convergence for an information need. To garner this understanding, the first thought might be to conduct user studies. However, user studies are expensive and allow users too much freedom which makes them impractical for fine-tuning models and analyzing the impact of specific interaction strategy. To evaluate URF systems with an emphasis on understanding the impact of interaction strategies, an automated approach is necessary.

This paper uses such an automated approach to analyze the impact of different interaction strategies for URF systems. First, we define several strategies for labeling positive and negative examples and study their impact on result quality. Second, we explore the impact of 4 classes of artificial users applying filters based on their level of domain knowledge. To evaluate the interaction strategies, we define automated evaluation protocols based on three multimedia collections, two from the interactive search challenges Lifelog Search Challenge [10] and Video Browser Showdown [28], and one for VOPE-8hr [39], a domain-specific forensic research collection. Figure 1 outlines the proposed evaluation process, where artificial users use the strategies to guide the relevance feedback process.

The knowledge gained from analyzing these interaction strategies can benefit the training of new users of URF systems, as well as help experienced users improve their performance on specific tasks. It could also be incorporated into systems as a feature to automatically suggest suitable interaction strategies for different tasks. Specifically, this paper contributes three best practice guidelines:

(1) Labeling strategies impact results significantly, in particular, strategies with more examples are not always better.
(2) Adding URF is always as good or better than just using filters.
(3) While filtering is often beneficial, overly aggressive filtering can adversely affect the ability to complete tasks.

2 RELATED WORK

User Relevance Feedback has been used since the 1960s to improve queries for information retrieval [24] and saw a boom in the 1990s and 2000s for multimedia retrieval [11, 12, 29, 33–35]. Later, it started fading as hash based approaches [20], product quantization [13], and deep learning models [7, 32, 40] were proven more efficient for retrieval on large-scale collections. However, in recent years the issue of scalability has largely been resolved and the state-of-the-art URF systems for large scale multimedia retrieval are competitive with other approaches and require fewer examples to train their models than the supervised approaches [16, 18, 36].

Since users are central to URF systems, it is important that the evaluation methods of these systems account for their behavior. Early evaluation efforts for relevance feedback utilized collections that had relevance judgement mappings between queries and associated documents [1, 2]. This allows for automating the evaluation process with the simulated “user” judging items based on the relevance judgement mappings. This form of evaluation with optimal users that have knowledge about the ground truth has remained the most common form for URF systems to date. Some evaluation protocols use this for labeling the suggestions as positive or negative [6, 29]. Other evaluation protocols, especially those that work with large-scale collections, also add additional arbitrary negatives [16, 23, 25, 36]. Analytic Quality uses artificial actors which solve an analytic task derived from an existing benchmark/user task, measuring precision and recall over time and estimating the user’s insight gain [38].

While the plethora of work on automated evaluations contributes to show the effectiveness of URF systems in various fields, the evaluation methodology only captures the behavior of a specific interaction strategy which may not be a strategy a real user will resort to. Aside from this there has also been work that has focused on evaluating systems with real users [23, 26, 29, 39]. URF systems typically showcase up to 30 images in each round and depending on the restrictions they can label as many items as they want [29], be limited to label a few examples [26], or only label examples as positive [25]. These evaluations give greater insight towards user behaviour, but they rarely generalize the interaction strategies due to the inherently limited set of users (tens at most).

With real users, it is also important to study the impact of users with different levels of knowledge. Dividing the users into users with relevant or no domain knowledge, it is possible to show that the performance of labeling examples or applying filters can be greatly affected [10, 27, 28].

From the related work, we identify a gap between artificial users and real users and to the best of our knowledge no work has focused on the impact this can have when evaluating URF systems. Hence there is a need for considering various labeling strategies that are inspired by real users, as well as establishing different levels of domain knowledge when applying filters.

3 USER INTERACTION STRATEGIES

To evaluate URF systems in an automated way, we need *artificial users*, software agents that simulate user behaviour. Their task is to find one or more relevant item(s) from a collection C, based on a textual description of an information need. To achieve this, they follow a certain strategy for labeling examples. Additionally the users apply filters based on different levels of domain knowledge. In the remainder of this section we propose a variety of labeling and filtering strategies to better understand the impact the different strategies can have on the performance of a URF system.

3.1 Labeling Strategies

The common labelling strategy of marking ground truth items as positives and everything else as negatives [4, 14, 16, 36] may result in a near empty set of positives and a vast set of negatives, where some negatives might feature relevant content. Furthermore, the task objective can involve finding all items in the relevant set, finding as many relevant items in r rounds as possible, or stopping once the first relevant item is encountered. To support evaluation for the different task objectives we must define strategies that use the ground truth items to rank suggestions and select the best suggestions as positives.

All strategies in this paper are based on observations of URF systems used in live interactive search challenges, in particular the

Lifelog Search Challenge and the Video Browser Showdown [10, 15, 17, 19, 21, 28]. We assume that the collections are comprised of images or videos represented with semantic feature vectors that can be compared using a distance metric. Each strategy uses a distance function with two feature vectors with semantic concepts extracted using neural networks as input; $d(\mathbf{v}_x, \mathbf{v}_{max})$. The first vector is the item from the suggestion set $\mathcal{S}_r$ of the current interaction round r. The second is the max-pooled feature vector of the relevant items. We use the Euclidean distance: it is simple, efficient, well-researched, and works well with using a compressed representation [36].[1]

We have identified three major categories of labeling strategies, *Accumulative Sets*, *Fixed Positive Sets* and *Arbitrary Negative Sets*, that we now describe in detail.

3.1.1 Accumulative Sets. Continuously adding items to the positive and negative set is a typical behavior of users that are attempting to gradually improve the model. This leads to the first strategy.

±*AccAdd* — Accumulative Sets with Additions: Label the p nearest items to $\mathbf{v}_{max}$ in $\mathcal{S}_r$ as positive, adding them to $\mathcal{P}_r$, the set of positives for round r, and label n furthest items from $\mathbf{v}_{max}$ in $\mathcal{S}_r$ as negative, adding them to $\mathcal{N}_r$, the set of negatives for round r.

$$\mathcal{P}_r = \mathcal{P}_{r-1} \cup \arg\min^{p}_{x \in \mathcal{S}_r} (d(\mathbf{v}_x, \mathbf{v}_{max})) \tag{1}$$

$$\mathcal{N}_r = \mathcal{N}_{r-1} \cup \arg\max^{n}_{x \in \mathcal{S}_r} (d(\mathbf{v}_x, \mathbf{v}_{max})) \tag{2}$$

As users keep adding to the sets, examples from earlier interaction rounds may become less important or even bad for the model. Therefore, users may start replacing items to improve the model; this behavior is typically observed from more experienced users.

±*AccRep* — Accumulative Set with Replacements: The user is allowed to replace items from the positive or negative sets if better representatives exist in the suggestion set $\mathcal{S}_r$, or in the labeled sets $\mathcal{P}_{r-1}$ and $\mathcal{N}_{r-1}$. The positive and negative sets at round r have the size pr and nr respectively.

$$\mathcal{P}_r = \arg\min^{pr}_{x \in \mathcal{S}_r \cup \mathcal{P}_{r-1} \cup \mathcal{N}_{r-1}} (d(\mathbf{v}_x, \mathbf{v}_{max})) \tag{3}$$

$$\mathcal{N}_r = \arg\max^{nr}_{x \in \mathcal{S}_r \cup \mathcal{P}_{r-1} \cup \mathcal{N}_{r-1}} (d(\mathbf{v}_x, \mathbf{v}_{max})) \tag{4}$$

An example of a positive example moving to the negative set is when an early positive example becomes a negative example as the model evolves. By replacing items, the chances of building a stronger model is enhanced. ±*AccRep* is a strategy that a user may utilize early on in a session but as the size of the positive and negative sets increases, the task of replacing items will become too time consuming. Therefore, even if this strategy works well, it may not be an optimal strategy during long sessions.

Enforcing an accumulative strategy increases the chances for overfitting the model towards certain features, which can be especially bad if erroneous items, e.g., incorrectly labeled, are added.

3.1.2 Fixed Positive Sets. Limiting the positive and negative sets to a fixed size, where the user can only replace items after the first round, avoids overwhelming the user with trying to replace from large sets. Instead, such strategies solely rely on the user's ability to replace bad examples when better ones are encountered. These strategies tend to be more dynamic, and are suitable for tasks where the user looks for strong archetypes to model the categories of relevance. However, limiting the sets can hurt the classification model for tasks where more items are required.

±*FixRep* — Fixed Set with Replacements: Restricts the size of both sets to p and n respectively.

$$\mathcal{P}_r = \arg\min^{p}_{x \in \mathcal{S}_r \cup \mathcal{P}_{r-1} \cup \mathcal{N}_{r-1}} (d(\mathbf{v}_x, \mathbf{v}_{max})) \tag{5}$$

$$\mathcal{N}_r = \arg\max^{n}_{x \in \mathcal{S}_r \cup \mathcal{P}_{r-1} \cup \mathcal{N}_{r-1}} (d(\mathbf{v}_x, \mathbf{v}_{max})) \tag{6}$$

Next is a hybrid strategy that users might use for extremely descriptive tasks, where good positive examples are rare, and where the limitation on negatives cannot train the model well enough to find the good positive examples. By restricting the positive set to only the strongest positives, but continuously adding to the negative set, the model can potentially be steered towards the relevant items. This strategy can be linked closely to negative relevance feedback as it is mainly guided by its negative set in the initial rounds [31].

+*FixRep-AccAdd* — Fixed Positive Set, Accumulative Neg. Set: Fixed positive set (Eq. 5) and accumulative negative set (Eq. 2).

3.1.3 Arbitrary Negative Sets. There is work that suggests that spending time on labeling negatives may not be as important as labeling positives [16, 38]. It is therefore crucial to investigate the impact of arbitrarily choosing negatives from either the suggested item set or the overall collection. Both use $arb(\mathcal{X}, n)$, a function which selects n arbitrary elements from a media item set $\mathcal{X}$. We define two strategies, whose positive strategies follow Eq. 1.

+*AccAdd-ArbLoc* — Arbitrary Negative Set (Local): Label n negatives arbitrarily from $\mathcal{S}_r$ and add them to $\mathcal{N}_{r-1}$.

$$\mathcal{N}_r = \mathcal{N}_{r-1} \cup arb(\mathcal{S}_r \setminus \mathcal{P}_r, n) \tag{7}$$

+*AccAdd-ArbGlo* — Arbitrary Negative Set (Global): Label n negatives from the entire collection C and add them to $\mathcal{N}_{r-1}$.

$$\mathcal{N}_r = \mathcal{N}_{r-1} \cup arb(C \setminus \mathcal{P}_r \setminus \mathcal{N}_{r-1}, n) \tag{8}$$

3.2 Filtering Strategies

Another aspect of interactive retrieval sessions is applying filters to reduce the scope of the analysis. Typically, filters are set based on metadata or features extracted from the items, and they can be added or removed at any point during a session. The reasoning behind the chosen filters can vary between users and the quality of filters can often depend on the level of *domain knowledge* they have. We define 4 types of users based on their expertise; *No Filter*, *Novice*, *Expert* and *Data Author.*

No Filters: To act as a baseline, this user type only utilizes the interactive learning system by labeling the retrieved suggestions, without applying any filters.

Novice: Users that tend to read a query and try to match the text with matching filters. This reflects the behavior of new users starting to work with a collection or system. However, it can also lead to misinterpretation of the query and exclusion of relevant items. This has indeed been observed during the novice sessions of interactive search challenges [10, 28]. The reasons for misinterpretations include time pressure, misusing the system, lack of domain knowledge, and language barriers. For example, consider the following

[1] We also experimented with Mahalanobis distance but found it to be less effective.

query: “Walking on a green footpath, to my car. I remember I had come off a flight and it was around lunch-time. I got into my car and drove to have a meal. No, I drove to work where I had lunch” [10]. A *Novice* user might attempt a filter such as “work”, excluding the relevant image in which a person was walking on a green footpath.

Expert: Represents analysts with expert domain knowledge. They know the system, have adequate knowledge of the collection and are able to interpret tasks beyond their description. They can connect query information with metadata from previous acquired knowledge, such as setting a location filter from a reference to a person who was seen at that location in a prior session. Semantic filters, such as "morning" or "evening" for hours, also fall under this.

Data Author: This type represents users with detailed knowledge, gained from having either created the collection or maintained it by updating or adding metadata. They are thus able to go beyond the query and apply filters due to actual recollection of creating the desired item in the collection. They may also use external information for clarification, as they may recall a detail which can be found via personal files or the internet, e.g., using a query regarding a place they visited but forgot the name of. Note that this user type is only applicable on collections that have a handful of contributors.

3.3 Summary

We conjecture that the labeling and filtering strategies presented in this section have a strong impact on the performance of URF systems. This has to date not been sufficiently captured by existing evaluation methods. In the remainder of the paper we therefore conduct experiments that analyze the impact of the different labeling and filtering strategies with a variety of collections and tasks.

4 EXPERIMENTAL SETUP

The experiments are conducted on tasks from 3 collections with varying objectives, query details and metadata quality. The key metric is completion time, or how many interaction rounds it takes on average to finish a task. While observing the behavior of labeling and filtering strategies for a handful of rounds is interesting, the direct impact of a strategy will ultimately be reflected in the time to finish the task. In addition, recall is also a relevant metric for tasks with time limits, or tasks that require finding all relevant items.

An actor [38] is an artificial user that uses a particular labeling and filtering strategy, and has a unique arbitrary starting point for each relevant task. Each actor communicates with a URF server using a script to perform the relevance feedback. To conduct the experiments we use a URF system where 25 items are suggested each round. The underlying classification model is linear SVM, which has a good accuracy/speed ratio and is consistent with the state of the art [16, 37]. During each interaction round the actor has to label p positives and n negatives from the 25 suggestions and apply filters depending on their labeling and filtering strategy respectively. All the results reported in this paper are an average of 50 different actors for each labeling and filtering strategy combination.

4.1 LifeLog Search Challenge 2019

Lifelogging is the idea of recording everything one does digitally, such as taking 2-3 images via a body camera every minute and logging daily routines manually and by using smart gadgets. Lifeloggers tend to end up with a large collection of images and metadata. The Lifelog Search Challenge (LSC) is an interactive live search challenge featuring a small curated lifelog collection [10]. The collection used in LSC2019 contains 41,666 images represented as 1000 dimensional feature vectors extracted with a deep neural network using concepts from ImageNet [5]. The collection contains metadata such as location, day and time that are useful as filters. Additional data, such as eating logs, fitness information, personal notes, are excluded as they are only available for a subset of the collection.

LSC2019 featured 24 interactive tasks with corresponding ground truths, where each task aimed to find images relevant to a textual query describing events from the lifelogger. The descriptions are extended through iterations, where each iteration adds some new information. Every iteration lasts 30 seconds with a total of six iterations. The nature of the task description is that of a memory, where one iteration may contradict a statement made in a previous one. The descriptions also typically contain information that can be correlated to metadata. The objective of a task is to find any of the relevant items; for some tasks the relevant set contains only a handful of images, while for a few it contains 50+ items. Due to the quality and transparency of the metadata, all types of users defined in Section 3.2 are applicable to this collection.

4.2 Video Browser Showdown 2020

The Video Browser Showdown [28] (VBS) is a live interactive search challenge similar to LSC. The latest edition of VBS was in 2020 and used the V3C1 collection that consists of 1000 hours of video segments or approximately 1M keyframes, from the online video site Vimeo [3]. We refer to this collection as VBS2020.

Unlike LSC2019, which consists of images from a single user, these videos are from different users all over the world. The users have free reign over the metadata, such as video categories and tags related to the video. While categories have a fixed number of options to select from, it is up to the user to determine which fits the video, making it highly subjective. The tags have no real restrictions, allowing the user to define their own tags. The categories and tags metadata are considered video level filters as they only refer to entire videos. The keyframes have also been processed for number of faces visible, making it possible to set this as a keyframe level filter. Note that our system uses keyframes from the videos as representatives of the segments. The representation for the keyframes is a more detailed feature vector with 12,988 dimensions [22].

VBS2020 featured 13 known item search tasks. The tasks describe visual events from a specific video segment. The descriptions focus more on visual features compared to metadata information. These tasks are presented through iterations as well, with time intervals of 60 seconds and a total of three iterations. The task objective is to find any relevant video segment of the described event. The collection also has more vaguely described tasks called Ad-hoc video search, where the goal is to find as many segments as possible that match the description. However, these tasks have been omitted since they lack a ground truth.

Due to the metadata having no central curator it is difficult to define a user for *Data Author* user strategy from Section 3.2. We therefore focus on *Novice* and *Expert* user strategies instead.

4.3 VOPE-8hr

The VOPE-8hr [30, 39] collection is inherently different from the previous two, both in terms of scale and objective. It consists of 8 hours of video broken into shots of 3 seconds, making it the smallest collection with ~9600 items. VOPE-8hr is a domain-specific collection for forensic research and the associated tasks are to find extremist propaganda content of three types; Neo-Nazis (task 1), Islamic terrorists (task 2) and Scottish ultra-nationalism (task 3). These tasks differ from the other two collections' tasks as the objective is to find *all* relevant examples of which some are easily identifiable and others being needles in a haystack. In addition to this, the collection is intentionally curated to have a portion of "red-herring" data that shares visual similarity to the relevant items but contextually is completely irrelevant [39]. The number of items to find for each task also differs, with tasks 1 and 3 having roughly 50 items and task 2 having 684 items. As there is no metadata that a real user could use as filters, no metadata filtering strategy experiments have been run for this collection.

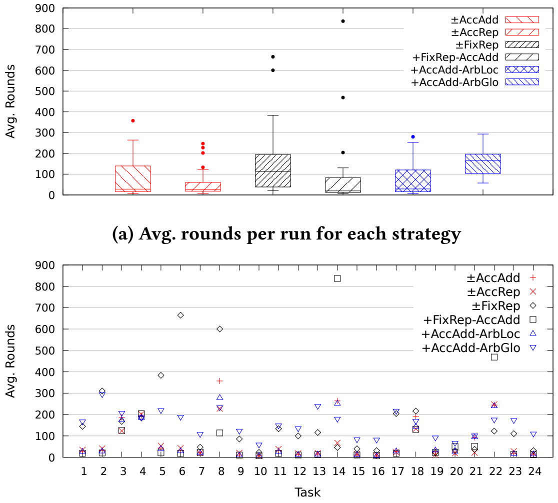
5 EXPERIMENT 1: LABELING STRATEGIES

A baseline experiment is run with all the labeling strategies from Section 3.1 where $p = 5$ and $n = 15$ with no filter options.[2] Typically in a URF setting the starting point is arbitrary. Therefore, selecting more negatives than positives in each round can be beneficial for directing the classifier quicker to the relevant item, as the initial items may not contain any good positive examples.

5.1 LSC2019

Figure 2(a) shows the number of rounds required to complete tasks for each labeling strategy on LSC2019. The two leftmost boxes show the distribution for the two *Accumulative* strategies. While their median is the same, ±*AccRep* is far more consistent than ±*AccAdd*, indicating that replacing items from the positive and negative set while adding items is better than just adding items. The two boxes in the middle show the *Fixed Positive* strategies. The hybrid strategy +*FixRep-AccAdd* is far better than the ±*FixRep* strategy. However, it does have some tasks where it does extremely poorly, as indicated by its outliers. We explore this in more detail with Figure 2(b) below. Lastly we have the two boxes on the right for the *Arbitrary Negative* results. Of the two, +*AccAdd-ArbLoc* is the better, meaning that labeling arbitrary negatives from the suggestion set is better than labeling them from the whole collection. If we compare this strategy with its *Accumulative* counterpart ±*AccAdd*, it is nearly identical in performance if not slightly better. Note that similar effect is observed from using -*ArbLoc* with +*FixRep* (not shown).

Figure 2(b) shows the average rounds per run for each task from LSC2019. The majority of tasks follow the pattern of tasks 1 and 2, which have ground truths with many near-duplicate images in the collection: "...looking at an old *clock*, with *flowers* visible. There was a *lamp* also..." (task 1), "A red *car* beside a white *house*..." (task 2). While most strategies fare well with these tasks, ±*FixRep* and +*AccAdd-ArbGlo* are consistently bad, but for different reasons. For these tasks, the ±*FixRep* strategy can end up in a state where it cannot improve the model as no stronger positive or negative examples can be found, and the strategy simply browses through the model's ranked list of suggestions. +*AccAdd-ArbGlo*, on the other hand, explores the search space more sporadically as it labels negatives from the whole collection. Since the actor must select some positives in every round, this sporadic exploration leads to many bad positives, resulting in a poor model for the tasks.



(a) Avg. rounds per run for each strategy

(b) Avg. rounds per run for each task

Figure 2: Baseline results for LSC2019. (a) shows the average rounds it takes to complete the tasks, emphasizing the distribution for each strategy. (b) shows the average rounds per run for each task separately.

There are a few tasks that show a different pattern. First, tasks 14 and 22 are cases where the description leads to many positive examples: "I was in my office taking a skype call... large image of a man's face on the *screen*..." (task 14). There is an abundance of computer, laptop, tablet, smartphone and tv related screens in this collection, and for this task the screen relates to a laptop/notebook screen which is seen in roughly one-third of the images in the collection. This can be bad for +*FixRep-AccAdd*, as the -*AccAdd* part will add many of these screens to its negative set. The other strategies counter this by adding screens also to the positive set. ±*FixRep* is a good option here, because of the exact same reason it is bad in the other tasks: few or none of these near-duplicates end up in the negative set, resulting in a better model. This scenario refutes the assumption that more examples are always better.

Second, task 8 has ground truth with visual features that are a mix of common and distinctive features, where the common features can lead the model away from the distinctive features: "Walking on a green footpath, to my *car*..." (task 8). Here, the ground truth item consists of a parking lot with several cars. While the collection consists of many different types of cars, some are abundant in the collection, while others are rarer. For this particular task, one of the abundant types is "minivan" which has approximately 3,000 related items, while one of the rarer types is "sports car" with roughly

[2] We have run experiments with $p \in \{1, 3, 5, 7\}$ and $n \in \{5, 10, 15, 20\}$. Since the results support the conclusions in this section, we do not discuss them further.

50 items. *+FixRep-AccAdd* is the best strategy in this case, as it limits the positive set to the strongest examples. While suggestions with the common “minivan” feature will continuously appear, some of them will be labelled negative, which in effect will allow the distinctive “sports car” feature to dominate the model.

5.2 VBS2020

Figure 3(a) shows the average rounds per run to complete the tasks. As this collection is roughly 20 times larger than LSC2019, we restrict the number of rounds a session can take to 500. Since some tasks cannot be completed within this limit, we also report the average recall distribution for each strategy in Figure 3(b).

Again, the two leftmost boxes in both figures represent the results for the *Accumulative* strategies. Here, ±*AccAdd* fares better than ±*AccRep*. With ±*AccRep* replacing items in growing positive and negative sets, the model can jump to many different directions, including back to areas already explored. This can occur when weak examples are repeatedly replaced with different weak examples. As the variety of content is much larger in the VBS2020 collection than LSC2019, this is more likely to happen.

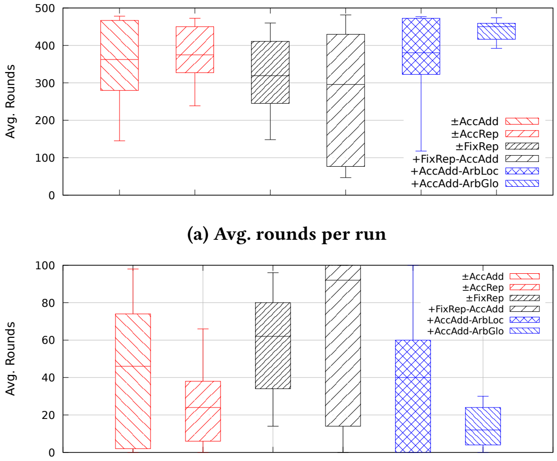
The middle two boxes show the results for the *Fixed Positive* strategies. For VBS2020, *+FixRep-AccAdd* is the best strategy for majority of the tasks in terms of rounds and recall. ±*FixRep* also shows great improvement over the other strategies, and has a more consistent distribution than *+FixRep-AccAdd* in terms of recall. The reason why these strategies do not fall into the same trap as ±*AccRep*, is because the positive set is limited, which allows for better control over the model’s direction. Even with ±*FixRep* replacing negatives, eventually only strong negative examples will remain.

The two rightmost boxes show the *Arbitrary Negative* strategies where the performance resembles that of LSC2019. *+AccAdd-ArbLoc* is again close to the performance of ±*AccAdd* as shown in the figure, and when tested with the *+FixRep* strategy (not shown) it resembles the performance of *+FixRep-AccAdd*. Based on these finding there is definitely some merit to letting the system choose arbitrary negatives from the suggestion set.

The objective of each task is to find any segment related to an event in a video, and as mentioned in Section 4.2 the task descriptions emphasize visual features of the ground truth items. The majority of the VBS2020 tasks have ground truth with a mixture of common and distinct visual features, similar to the outlier task 8 for LSC2019. As described above, many positives hurt the classification model with such tasks, as the positive set includes many items with strong common features, which drive the model away from the ground truth items with distinctive features. Policies with few positive examples are better candidates in this case. The prevalence of this type of tasks is the main reason for the strong performance of the ±*FixRep* and *+FixRep-AccAdd* strategies for VBS2020.

5.3 VOPE-8hr

The results are significantly different for the VOPE-8hr collection, as the focus of tasks is to find *all* relevant items. Figure 4(a) shows the number of average rounds with each labeling strategy for each task. The bottom points of each line depict the average round when the first relevant item was encountered and the upper point is the average rounds it took find all relevant items and complete the task.



(a) Avg. rounds per run

(b) Avg. recall per run

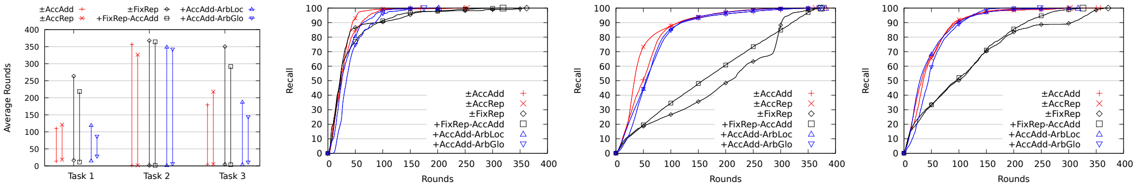
Figure 3: Baseline results for VBS2020. (a) shows the number of avg. rounds it takes to complete the tasks for each labeling strategy and (b) shows the average recall of them.

The red lines show the *Accumulative* strategies, where there are no significant differences between them. The black lines depict the *Fixed Positive* strategies, which for these tasks perform the worst. As the intention of the tasks is to find all relevant items, the number of examples in the positive set may be too small to define a satisfactory model. The blue lines show the *Arbitrary Negative* strategies. Here, we observe that *+AccAdd-ArbGlo* does surprisingly well for all 3 tasks. This is due to the presence of "red-herring" data along with noise in the collection, making it difficult to select good negatives from the local suggestion set. Task 2 takes the longest, as it has 14 times as many relevant items as the other two tasks.

If the objective was to find the first relevant item, all strategies are efficient. This is more clear when looking at the recall over rounds for each task, depicted in Figures 4(b)-(d) for each task respectively. For task 1, all strategies find more than 80% of the relevant items in fewer than 50 rounds but struggle with the remaining 20%. While most strategies follow a similar pattern for tasks 2 and 3, the *Fixed Positive* strategies finds the majority at a slower rate but complete the task at the same time as the others. This behavior can be related to the "red herring" data as the other strategies add far more of those into their positive set which leads their models quicker to the relevant search space. Ultimately, it is worth considering that when such a scenario occurs where the user goes many rounds without discovering a relevant item, it could indicate to the system that it should guide the user to switching strategies.

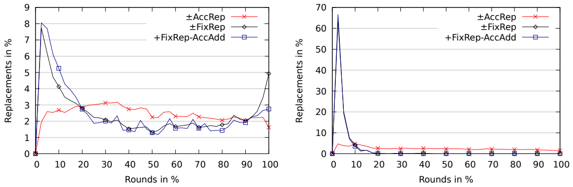
5.4 Analysis of Replacements

Since some strategies allow replacements, it is interesting to know when those replacements occur. Figure 5(a) shows the replacement occurrences for ±*AccRep*, ±*FixRep* and *+FixRep-AccAdd* for LSC2019.



(a) **(b)** **(c)** **(d)**

Figure 4: Results from baseline labeling strategy experiments for VOPE-8hr. (a) shows the number of avg. rounds for each labeling strategies and task. The bottom point indicates the avg. round the first relevant item was discovered, while the top depicts the avg. rounds it took to complete the task. (b), (c) and (d) shows the average recall over rounds for each task respectively.



(a) LSC2019 **(b) VOPE-8hr**

Figure 5: Occurence of replacements in the positive set.

The y-axis shows the average replacement occurrences as percentage and the x-axis shows the rounds as percentage. For the *Fixed Positive* strategies, many replacements occur early in the session. This is expected, as the model is starting to form and every round will have different suggestions. As the model becomes better, however, replacements become rare as many of the suggestions are similar and not necessarily better. For ±*AccRep* the occurrence of replacements is more balanced; as the positive set keeps increasing, the chance of replacements occurring remains similar.

The replacement patterns are similar across all collections. The replacement pattern for VBS2020 (not shown) is nearly identical to LSC2019. Figure 5(b) shows the replacement occurrences for VOPE-8hr which has a similar pattern but the trend of the fixed strategies is far more apparent, where both strategies stop replacing items after roughly 20% into the session. This is because the “red-herring” data in the collection is visually similar to the relevant items, helping the system to rapidly find optimal positive examples.

5.5 Summary

From the labeling strategy experiments we learn that different strategies can be beneficial depending on the collections content and size, and the nature of the tasks. This is a clear indication that the current evaluation methods that use strategies resembling *+AccAdd-ArbGlo* are not good enough to indicate the quality of URF systems. In addition to this revelation, the results contradict the assumption that more positive/negative labeled examples each round always lead to faster convergence.

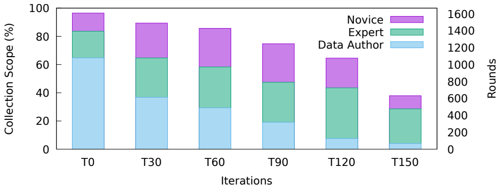


Figure 6: Average scope of the tasks from LSC2019 when their filters are applied, shown with percentage (left axis) and maximum number of interaction rounds (right axis) when the number of suggestions per round is 25.

6 EXPERIMENT 2: FILTERING STRATEGIES

We have observed how different labeling strategies can impact the number of rounds it takes to solve tasks for the different datasets. On average the VBS2020 and VOPE-8hr take roughly 230 rounds to complete a task, which tranlates to 75 minutes assuming the user spends an average of 20 seconds judging examples per interaction round. LSC2019 tasks fare better with average tasks for the strongest strategies taking fewer than 75 rounds (25 minutes). However, considering the actual time to complete the tasks in the live search challenges is 5-7 minutes, this is too long. In this experiment we run the best labeling strategies with the different filtering strategies described in Section 3.2 on LSC2019 and VBS2020. As previously mentioned VOPE-8hr does not contain metadata that can act as filters and therefore experiments for it have been omitted.

6.1 LSC2019

We start by analyzing the potential impact of filters. Figure 6 shows the percentage of search space when filters are applied by the 3 different user types for each iteration of the tasks in LSC2019. Furthermore, it depicts the worst case number of rounds it will take to find the relevant items on the right axis.

Overall this indicates the possibility of faster retrieval with the scope being reduced by more than 60% when all filters are applied for the *Novice* user. Additionally the relation between type of user and scope is clear and shows that in the worst case *Data Author* need much fewer rounds than the *Expert*, while the difference between *Expert* and *Novice* is slightly smaller. Note that the *Novice* users apply filters that exclude the relevant items for four tasks, meaning

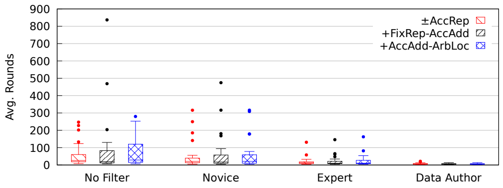


Figure 7: Avg. rounds to complete tasks with filters using the best labeling strategies for LSC2019.

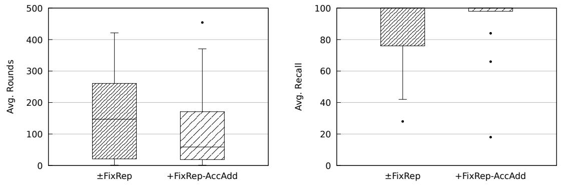
that the retrieval process either stops by finding the relevant item in an iteration prior to the one where the excluding filters are applied or they run out of items once they are applied.

Turning to the actual impact of filters, Figure 7 shows the results for actors using the best strategy from each category of the baseline experiments, ±*AccRep*, *+FixRep-AccAdd* and *+AccAdd-ArbLoc*, with regards to average number of rounds per run to complete the task when the different filtering strategies are applied. The leftmost group of boxes represents the same results from Figure 2(a) where no filters were applied. The next group is the results from running the filters applied by the *Novice* which sees each strategy taking fewer than 70 rounds in average across tasks. The results include the four failure tasks which are shown as outliers.[3] The third group of boxes represent the results where the filters are applied by an *Expert*. Again, this shows great improvement for all strategies with average rounds being between 20-30 for all strategies. The final group of boxes show the result for the *Data Author* filters, which brings all strategies below 10 rounds. Overall the trend is expected, as users with better domain knowledge apply better filters and avoid exclusion issues. As a final note, we observe no change in the relative performance between labeling strategies when filters are applied: ±*AccRep* and *+FixRep-AccAdd* remain the strongest.

6.2 VBS2020

For VBS2020, the *Novice* selects most filters from tags and categories which end up excluding the correct video segment for most tasks. In fact there are only two tasks where it manages to not exclude them and manages to find the desired segment in fewer than 20 rounds. For the remaining 11 tasks, however, it fails to complete any of them with the filters applied. We therefore do not consider the *Novice* user further.

Next we study the impact of actors using the filtering strategy of an *Expert* user that has worked with the collection and understands when and how to set frame level filters and video level filters. Figure 8(a) highlights the results of these actors using the *Fixed Positive* strategies, which were the best overall from the labeling strategy experiments. None of the filters set by the actor exclude any relevant items and we see a great improvement in terms of average rounds per run for both strategies with *+FixRep-AccAdd* still being the best. Figure 8(b) shows the avg. recall per run for the actors. While both labeling strategies improve in this area as well, the *+FixRep-AccAdd* is far more consistent with the average recall close to 100% for the majority of the tasks, which further solidifies it as a preferred strategy. However, there are still tasks where some of the runs fail to complete. This means that the model either got derailed and exceeded the number of rounds or that even with the filters applied the search scope is still too large.



(a) Average rounds per run

(b) Average Recall

Figure 8: VBS2020 results for the *Fixed* strategies using *Expert* filters.

6.3 Summary

Overall we have shown that applying filters is beneficial for all types of users if the collection is well curated and the task descriptions reflect the metadata used as filters. It can have negative consequences if the user has little domain knowledge, especially when they set extreme filters that exclude the relevant items. However, as filters are set over time and the excluding feature is not set at the beginning, URF is occasionally fast enough to bypass the excluding filter by finding a relevant item before that filter is set. Furthermore, our results firmly indicate that URF with filters applied by users with high domain knowledge is always better than just applying filters.

7 CONCLUSION

In this paper, we have analyzed the impact of interaction strategies for labeling positives and negatives as well as applying filters based on user's domain knowledge for user relevance feedback systems. By conducting experiments on three different collections of various sizes and tasks using artificial users, we observe that the choice of labeling strategy can have a major impact on number of interaction rounds it takes to finish a task. There is no single optimal labeling strategy, as the best strategy depends on both the collection and the task. Furthermore, our results refute the common assumption of providing more training examples is always beneficial, as strategies with smaller set of examples lead to better results in some cases.

We observe that users with expert level or higher domain knowledge unsurprisingly apply filters that are beneficial. However, aggressive filtering, especially by novice users, can hinder the completion of tasks. Furthermore, URF is a powerful tool in conjunction with filters that leads to better results than using filters alone.

These findings should be considered in future URF evaluation efforts as more refined artificial users will lead to better benchmarks, making it easier to quantify the performance of URF systems.

ACKNOWLEDGMENTS

This work was supported by a PhD grant from the IT University of Copenhagen and by the European Regional Development Fund (project Robotics for Industry 4.0, CZ.02.1.01/0.0/0.0/15 003/0000470).

[3] Note that ±*AccRep* does complete up to 11 of its 50 runs for 1 of those tasks. This is due to the excluding filter being set later for this task than the other 3, making it possible for the task to be completed.

REFERENCES

[1] IJsbrand Jan Aalbersberg. 1992. Incremental Relevance Feedback. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Copenhagen, Denmark) *(SIGIR '92)*. Association for Computing Machinery, New York, NY, USA, 11–22.

[2] James Allan. 1996. Incremental Relevance Feedback for Information Filtering. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Zurich, Switzerland) *(SIGIR '96)*. Association for Computing Machinery, New York, NY, USA, 270–278.

[3] Fabian Berns, Luca Rossetto, Klaus Schoeffmann, Christian Beecks, and George Awad. 2019. V3C1 Dataset: An Evaluation of Content Characteristics. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval* (Ottawa ON, Canada) *(ICMR '19)*. Association for Computing Machinery, New York, NY, USA, 334–338.

[4] Bogdan Boteanu, Ionuţ Mironică, and Bogdan Ionescu. 2017. Pseudo-relevance feedback diversification of social image retrieval results. *Multimedia Tools and Applications* 76, 9 (2017), 11889–11916.

[5] François Chollet. 2017. Xception: Deep Learning with Depthwise Separable Convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Honolulu, HI, USA, 1800–1807.

[6] Duc-Tien Dang-Nguyen, Luca Piras, Giorgio Giacinto, Giulia Boato, and Francesco GB DE Natale. 2017. Multimodal retrieval with diversification and relevance feedback for tourist attraction images. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 13, 4 (2017), 1–24.

[7] Lianli Gao, Jingkuan Song, Fuhao Zou, Dongxiang Zhang, and Jie Shao. 2015. Scalable Multimedia Retrieval by Deep Learning Hashing with Relative Similarity Learning. In *Proceedings of the 23rd ACM International Conference on Multimedia* (Brisbane, Australia) *(MM'15)*. Association for Computing Machinery, New York, NY, USA, 903–906.

[8] Ralph Gasser, Luca Rossetto, and Heiko Schuldt. 2019. Multimodal Multimedia Retrieval with Vitrivr. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval* (Ottawa ON, Canada) *(ICMR '19)*. Association for Computing Machinery, New York, NY, USA, 391–394.

[9] Paula Gómez Duran, Eva Mohedano, Kevin McGuinness, Xavier Giró-i Nieto, and Noel E. O'Connor. 2018. Demonstration of an Open Source Framework for Qualitative Evaluation of CBIR Systems. In *Proceedings of the 26th ACM International Conference on Multimedia* (Seoul, Republic of Korea) *(MM'18)*. Association for Computing Machinery, New York, NY, USA, 1256–1257.

[10] Cathal Gurrin, Klaus Schoeffmann, Hideo Joho, Andreas Leibetseder, Liting Zhou, Aaron Duane, Dang Nguyen, Duc Tien, Michael Riegler, Luca Piras, et al. 2019. Comparing approaches to interactive lifelog search at the lifelog search challenge (LSC2018). *ITE Transactions on Media Technology and Applications* 7, 2 (2019), 46–59.

[11] Xiaofei He, Oliver King, Wei-Ying Ma, Mingjing Li, and Hong-Jiang Zhang. 2003. Learning a semantic space from user's relevance feedback for image retrieval. *IEEE transactions on Circuits and Systems for Video technology* 13, 1 (2003), 39–48.

[12] Thomas S Huang, Charlie K Dagli, Shyamsundar Rajaram, Edward Y Chang, Michael I Mandel, Graham E Poliner, and Daniel PW Ellis. 2008. Active learning for interactive multimedia retrieval. *Proc. IEEE* 96, 4 (2008), 648–667.

[13] Herve Jegou, Matthijs Douze, and Cordelia Schmid. 2010. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence* 33, 1 (2010), 117–128.

[14] Lu Jiang, Teruko Mitamura, Shoou-I Yu, and Alexander G. Hauptmann. 2014. Zero-Example Event Search Using MultiModal Pseudo Relevance Feedback. In *Proceedings of International Conference on Multimedia Retrieval* (Glasgow, United Kingdom) *(ICMR '14)*. Association for Computing Machinery, New York, NY, USA, 297–304.

[15] Björn Þór Jónsson, Omar Shahbaz Khan, Dennis C. Koelma, Stevan Rudinac, Marcel Worring, and Jan Zahálka. 2020. Exquisitor at the Video Browser Showdown 2020. In *MultiMedia Modeling*, Yong Man Ro, Wen-Huang Cheng, Junmo Kim, Wei-Ta Chu, Peng Cui, Jung-Woo Choi, Min-Chun Hu, and Wesley De Neve (Eds.). Springer International Publishing, Cham, 796–802.

[16] Omar Shahbaz Khan, Björn Þór Jónsson, Stevan Rudinac, Jan Zahálka, Hanna Ragnarsdóttir, Þórhildur Þorleiksdóttir, Gylfi Þór Guðmundsson, Laurent Amsaleg, and Marcel Worring. 2020. Interactive Learning for Multimedia at Large. In *European Conference on Information Retrieval*. Springer, Cham, 495–510.

[17] Omar Shahbaz Khan, Mathias Dybkjær Larsen, Liam Alex Sonto Poulsen, Björn Þór Jónsson, Jan Zahálka, Stevan Rudinac, Dennis Koelma, and Marcel Worring. 2020. Exquisitor at the Lifelog Search Challenge 2020. In *Proceedings of the Third Annual Workshop on Lifelog Search Challenge* (Dublin, Ireland) *(LSC '20)*. Association for Computing Machinery, New York, NY, USA, 19–22.

[18] Miroslav Kratochvil, František Mejzlík, Patrik Veselý, Tomáš Soućek, and Jakub Lokoć. 2020. *SOMHunter: Lightweight Video Search System with SOM-Guided Relevance Feedback*. Association for Computing Machinery, New York, NY, USA, 4481–4484.

[19] Miroslav Kratochvíl, Patrik Veselý, František Mejzlík, and Jakub Lokoč. 2020. SOM-Hunter: Video Browsing with Relevance-to-SOM Feedback Loop. In *MultiMedia Modeling*. Springer International Publishing, Cham, 790–795.

[20] Brian Kulis and Kristen Grauman. 2009. Kernelized Locality-Sensitive Hashing for Scalable Image Search. In *2009 IEEE 12th International Conference on Computer Vision*. IEEE Computer Society, Los Alamitos, CA, USA, 2130 – 2137.

[21] František Mejzlík, Patrik Veselý, Miroslav Kratochvíl, Tomáš Souček, and Jakub Lokoč. 2020. SOMHunter for Lifelog Search. In *Proceedings of the Third Annual Workshop on Lifelog Search Challenge* (Dublin, Ireland) *(LSC '20)*. Association for Computing Machinery, New York, NY, USA, 73–75.

[22] Pascal Mettes, Dennis C. Koelma, and Cees G.M. Snoek. 2016. The ImageNet Shuffle: Reorganized Pre-Training for Video Event Detection. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval (ICMR '16)*. Association for Computing Machinery, New York, NY, USA, 175–182.

[23] G. L. J. Pingen, M. H. T. de Boer, and R. B. N. Aly. 2017. Rocchio-Based Relevance Feedback in Video Event Retrieval. In *MultiMedia Modeling*. Springer International Publishing, Cham, 318–330.

[24] J. J. Rocchio. 1965. *Relevance feedback in information retrieval.* Technical Report. University of Harvard, Computer Laboratory.

[25] Ork De Rooij and Marcel Worring. 2012. Efficient Targeted Search Using a Focus and Context Video Browser. *ACM Trans. Multimedia Comput. Commun. Appl.* 8, 4, Article 51 (Nov. 2012), 19 pages.

[26] Yong Rui, Thomas S Huang, and Sharad Mehrotra. 1997. Content-based image retrieval with relevance feedback in MARS. In *Proceedings of International Conference on Image Processing*, Vol. 2. IEEE, IEEE, Santa Barbara, CA, USA, 815–818.

[27] Sheikh Muhammad Sarwar, John Foley, and James Allan. 2018. Term Relevance Feedback for Contextual Named Entity Retrieval. In *Proceedings of the 2018 Conference on Human Information Interaction & Retrieval* (New Brunswick, NJ, USA) *(CHIIR '18)*. Association for Computing Machinery, New York, NY, USA, 301–304.

[28] Klaus Schoeffmann. 2014. A User-Centric Media Retrieval Competition: The Video Browser Showdown 2012-2014. *IEEE MM* 21, 4 (2014), 8–13.

[29] Roberto Tronci, Gabriele Murgia, Maurizio Pili, Luca Piras, and Giorgio Giacinto. 2013. *ImageHunter: A Novel Tool for Relevance Feedback in Content Based Image Retrieval.* Springer Berlin Heidelberg, Berlin, Heidelberg, 53–70.

[30] VOX-Pol. 2021. About Us. https://www.voxpol.eu/about-us

[31] Xuanhui Wang, Hui Fang, and ChengXiang Zhai. 2008. A Study of Methods for Negative Relevance Feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Singapore, Singapore) *(SIGIR '08)*. Association for Computing Machinery, New York, NY, USA, 219–226.

[32] Yunchao Wei, Yao Zhao, Canyi Lu, Shikui Wei, Luoqi Liu, Zhenfeng Zhu, and Shuicheng Yan. 2016. Cross-modal retrieval with CNN visual features: A new baseline. *IEEE transactions on cybernetics* 47, 2 (2016), 449–460.

[33] Rong Yan, Alexander Hauptmann, and Rong Jin. 2003. Multimedia Search with Pseudo-relevance Feedback. In *Image and Video Retrieval*. Springer Berlin Heidelberg, Berlin, Heidelberg, 238–247.

[34] Rong Yan, Alexander G. Hauptmann, and Rong Jin. 2003. Negative Pseudo-Relevance Feedback in Content-Based Video Retrieval. In *Proceedings of the Eleventh ACM International Conference on Multimedia* (Berkeley, CA, USA) *(MULTIMEDIA '03)*. Association for Computing Machinery, New York, NY, USA, 343–346.

[35] Yi Yang, Feiping Nie, Dong Xu, Jiebo Luo, Yueting Zhuang, and Yunhe Pan. 2011. A multimedia retrieval framework based on semi-supervised ranking and relevance feedback. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 4 (2011), 723–742.

[36] Jan Zahálka, Stevan Rudinac, Björn Þór Jónsson, Dennis C Koelma, and Marcel Worring. 2018. Blackthorn: Large-Scale Interactive Multimodal Learning. *IEEE Transactions on Multimedia* 20, 3 (2018), 687–698.

[37] Jan Zahálka, Stevan Rudinac, Björn Þór Jónsson, Dennis C. Koelma, and Marcel Worring. 2016. Interactive Multimodal Learning on 100 Million Images. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval* (New York, New York, USA) *(ICMR '16)*. Association for Computing Machinery, New York, NY, USA, 333–337.

[38] Jan Zahálka, Stevan Rudinac, and Marcel Worring. 2015. Analytic Quality: Evaluation of Performance and Insight in Multimedia Collection Analysis. In *Proceedings of the 23rd ACM International Conference on Multimedia* (Brisbane, Australia) *(MM '15)*. Association for Computing Machinery, New York, NY, USA, 231–240.

[39] Jan Zahálka, Marcel Worring, and Jarke J. Van Wijk. 2021. II-20: Intelligent and pragmatic analytic categorization of image collections. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2021), 422–431.

[40] Shifeng Zhang, Jianmin Li, Mengqing Jiang, Peijiang Yuan, and Bo Zhang. 2017. Scalable discrete supervised multimedia hash learning with clustering. *IEEE Transactions on Circuits and Systems for Video Technology* 28, 10 (2017), 2716–2729.

Exquisitor: Responsive, Accurate, Flexible and Scalable Interactive Learning for Multimedia

Omar Shahbaz Khan*, Björn Þór Jónsson*, Jan Zahálka†, Stevan Rudinac‡, Marcel Worring‡
*IT University of Copenhagen
†Czech Technical University in Prague
‡University of Amsterdam

***Abstract*—Interactive multimedia retrieval needs to be responsive, accurate, flexible and scalable to deal with continuously evolving user information needs in large collections. These requirements are hampered by three fundamental gaps between human and machine, namely the semantic, pragmatic and scale gap. While approaches exist that alleviate one or two of the gaps, no approach exists that bridges them all simultaneously. We propose Exquisitor, a novel interactive learning approach that bridges all three gaps, featuring a new cluster-based retrieval mechanism, combining high-dimensional indexing, incremental retrieval and query optimisation policies. Using these techniques, Exquisitor facilitates search and exploration in collections with over 100 million items, while maintaining sub-second latency. Our experiments establish Exquisitor as the state-of-the-art approach satisfying all requirements expected from a truly interactive multimedia retrieval system.**

***Index Terms*—Interactive multimodal learning, multimedia analytics, high-dimensional indexing, incremental retrieval, query optimisation.**

I. Introduction

Multimedia collections are an integral part of everyday life, as well as a crucial source of data for science, public health, entertainment, and more. With this vast amount of multimedia data available, users need approaches facilitating search and exploration in their collections. Especially with the emergence of smartphones, wireless networking and cloud services, users have become accustomed to **responsive** applications. Recently, 0.1 to 2-3 seconds has been established as an acceptable latency for tasks involving interaction with the user [34].

User interaction with multimedia data is dominated by three gaps, represented in Figure 1. The first challenge is the machine's capability to produce semantically relevant results. This is described as the *semantic gap*, which revolves around the semantic level of information extracted from a multimedia item by human and machine [41]. The user can recognise content and assert context almost instantly. The machine, on the other hand, bases its understanding on objective concepts, extracted from annotations and low-level features. The last decade's research in deep learning has greatly reduced the semantic gap, with Convolutional Neural Networks approaching or surpassing human capabilities in tasks such as object recognition [24], [42]. While moving from benchmarks to general applications still remains an open challenge [3], the state of the art already provides **accurate** results.

With accurate results, the machine can start addressing the user's intent, a set of information needs that dictate interactions with a collection [22]. The user's intent is dynamic, involving a complex interplay of content and context that is generally encoded in the machine's concept dictionary only partially. This is reflected in the *pragmatic* gap, which is the difference between the highly adaptive model in the human mind that creates, adds and deletes categories on the fly, as opposed to the machine which is typically limited to rigid and non-adaptive models [50]. To address the pragmatic gap, the machine needs a **flexible** model that matches the adaptivity of the human mental model.

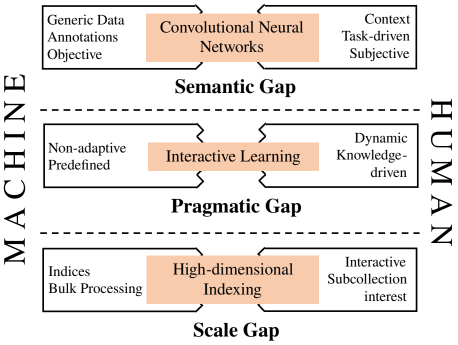


Fig. 1: The three fundamental gaps between human and machine and the approaches to help bridge them. Exquisitor is the first multimedia system to simultaneously bridge all three.

Interactive learning (IL) is a key method for bridging the pragmatic gap, as it solves analytic tasks that require alternating between exploration and search [50]. IL is a human-in-the-loop machine learning approach, where a user judges the relevance of a collection subset by labeling the media items as positive or negative. The judged items are used to (re)train an underlying classifier in the system to find a new subset of items to present the user with [16]. Recently, interactive learning has been deployed in a variety of settings, such as deep learning with user relevance feedback [33], technology-assisted reviewing using continuous active learning [27], multimedia analytics [51], or interactive multimodal learning in compressed domain [49].

On top of the accuracy and flexibility challenges, the continuous growth of multimedia collections brings the challenge

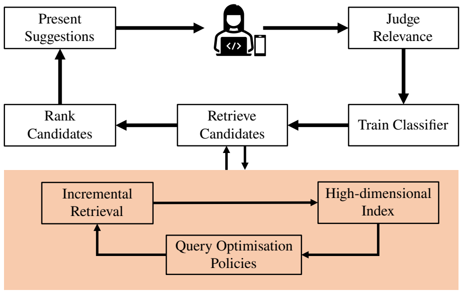


Fig. 2: The Exquisitor approach. The feedback loop represents state of the art elements in Interactive Learning. The highlighted elements are those innovated on by Exquisitor.

of scale. Increasing computational resources can alleviate this problem, but cannot solve it. In addition, the amount of items a user and a machine can process at once differs immensely. Where the machine can index the data and process in bulk, the human can only process a small subset to avoid being overwhelmed. Thus the user requires the ability to dynamically define and explore subcollections rather than the entire collection. This constitutes the *scale gap*. A **scalable** approach is capable of working with such large-scale collections [19].

When users do not have a clear information need or are unable to formulate it as a query, exploratory operations are better for gaining insight. As the user gains insight, the information need may become clear enough to seek specific items. This interaction is dictated by the user, where the actions frequently switch between (sub)collection exploration and search [50]. To provide such interactions, an interactive retrieval system is needed that has all four aforementioned characteristics: responsiveness, accuracy, flexibility and scalability.

Available state-of-the-art techniques, such as deep learning for retrieval, hashing, or self-organising maps, are responsive and accurate, but they are not flexible as they are limited to their precomputed similarity structure. Some are scalable in principle, but are rarely evaluated on large-scale collections [52]. Scalable approaches reduce computations, typically by applying approximate high-dimensional indexing on large-scale multimedia collections, which can greatly reduce the amortized computation cost. This makes such approaches responsive, accurate and scalable, but not flexible, as they revolve around the precomputed distribution of the data.

We propose Exquisitor, a responsive, accurate, flexible and scalable interactive learning approach for exploration and search in multimedia collections. Exquisitor uses limited hardware resources, making it available to broad audiences. Figure 2 depicts Exquisitor's innovation within the framework of interactive learning, which involves:

1) High-Dimensional Index: This component is at the core of Exquisitor. Unlike most existing scalable approaches, the index is optimised for supporting interactive learning.
2) Incremental Retrieval: In many exploration and search scenarios, traditional use of the high-dimensional index fails to return accurate results. Exquisitor applies incremental retrieval to ensure processing until the relevant items are found.
3) Query Optimisation Policies: Exquisitor uses information about data distribution with regards to the query and the interactive session to enable incremental retrieval early in the process.

Through benchmarking against various baselines using collections ranging from 40K to over 100M items in size, we show that Exquisitor is capable of supporting both interactive exploration and search tasks, while maintaining sub-second latency using a single CPU core, making Exquisitor the state-of-the-art large-scale multimodal interactive learning approach.

II. Related Work

In this section we describe the state of the art in interactive learning to identify advantages and limitations. Based on this we provide a set of requirements for high-dimensional indexing to facilitate interactivity on extremely large collections. These requirements are then used to reflect upon the state of the art in high-dimensional indexing.

A. Interactive Learning

Interactive learning has been an integral part of content-based retrieval from its dawn [37]. There are two main types of IL approaches, User Relevance Feedback (URF) and Active Learning (AL) [16]. In recent years, URF and AL approaches have shown successful incorporation of human (expert) knowledge to provide exploratory access to ever-growing collections. Furthermore, they have been able to learn new analytic categories [49], [23] and train accurate classifier with minimal number of training samples [47].

The key difference between URF and AL is the returned data subset from the system. In URF the subset consists of items that the classifier is most confident about [37]. AL, on the other hand, presents the user with items that the classifier is least certain about in order to maximize the information gain of the classifier [16], [17]. In the recent years batch active learning has gained popularity in deep computer vision applications [28], [48]. However, it requires human users to label a large set of items first, which makes them non-interactive. There has been research in stopping strategies for AL, to make faster use of the classifier [27], but these still require a fair amount of labeled items and lack flexibility. With user in the loop, URF is the preferred approach, as it satisfies the need to find relevant items during each interaction round and also makes it easier for the user to judge the items.

There are two phases in an IL system for multimedia retrieval, the offline and online phase. The offline phase performs precomputations that expedite the interactive analysis in the online phase. This includes extracting various semantic features, which are used to represent each item. In the interactive context, using features that transparently convey semantics to the user has been shown to diminish the semantic gap and lead to a more accurate classifier [50], [41]. For efficiency in larger collections it is typical to compress these features [49], [9]. The final data representation can be used as is or be stored in a data structure to help with the search process, such as a high-dimensional index [4].

The non-highlighted elements in Figure 2 depict the online phase of the IL process. The process starts by presenting the user with a small arbitrary subset of the collection, typically containing around 20-30 items [46]. The small size ensures the user is not overwhelmed and the classifier remains adaptable in the early rounds. The user judges these presented items by labeling them as positive or negative [14]. The system may constrain how many items need to be judged [37] or it may not [46]. The labels are used to update the interactive classifier and query the collection. The top-ranked items are then presented to the user in the new interaction round.

On smaller collections, many relevance feedback methods have been used for content based retrieval, such as deep learning where URF is used to fine-tune a CNN [33] and SVM with various feature representations [14]. To the best of our knowledge, only two approaches have been proven successful on very large collections, such as YFCC100M with 100M items [45]. The first is based on Product Quantization [18], which is a popular approach for k-NN search using a compression scheme that splits the high-dimensional features into low-dimensional sub-spaces. The second approach is Blackthorn [49], which also uses a compressed representation, but the representation preserves original values from the most important features. Blackthorn has proven capable of achieving average interaction round response times of 1.2 seconds on YFCC100M using 16 CPU cores, while outperforming product quantization in terms of accuracy. However, Blackthorn relies on many CPU-cores to be scalable, making it infeasible to process large-scale collections with modest hardware. With this in mind we assume Blackthorn to be the main baseline for our use case.

B. Indexing

To improve the efficiency of URF for large collections, storing the data in a high-performance index is a natural extension as it structures the data for faster retrieval. We identify three main requirements for an index to enhance large-scale URF.

R1 *Short and Stable Response Time:* A successful indexing approach in interactive learning combines good result quality with response time guarantees [44].

R2 *Preservation of Feature Space Similarity Structure:* The space partitioning of the high-dimensional indexing algorithm must preserve the similarity structure on the feature space used by the interactive classifiers.

R3 *k Farthest Neighbours:* Relevance feedback approaches present the most confident relevant items based on the judgments observed so far, which are the items farthest from the classification boundary.

Due to the curse of dimensionality, scalable high-dimensional indexing methods must rely on approximation using scalar or vector quantization, typically trading off small reductions in quality (or even just quality guarantees) for dramatic response time improvements. Therefore, when employing an approximate high-dimensional index there is a major focus on the trade-off between quality and time.

The popular hash-based index LSH uses random projections acting as locality preserving hashing functions [10], [1], storing the data in buckets within tables. As the quality and performance of LSH is highly dependent on the hash functions, some approaches focus on improving these [43], [32] while others focus on reducing the amount of functions [30]. LSH primarily focuses on quality over performance, failing **R1**, as well as "slicing" the high-dimensional space leading to difficulties when ranking based on distance, thus failing **R2**. To the best of our knowledge LSH has not been considered in the context of *hyperplane-based farthest-neighbour* queries, thus having no guarantee for **R3**.

Vector quantization typically uses clustering approaches, such as k-means, to determine a set of representative feature vectors to use for the quantization. They fail to satisfy **R1** as they typically end with a large portion of the collection in a few clusters or even a single cluster. By relying on Voronoï cells in the high-dimensional space, they satisfy **R2**. As they can store the entire features, they can rank the results from the farthest clusters, satisfying **R3** [12].

Product quantization (PQ) and its variants [18], [2] cluster the high-dimensional vectors into low-dimensional sub-spaces that are indexed independently. As PQ modifies the space, it fails to satisfy **R2** on top of the already existing failure to satisfy **R1**. The extended Cluster Pruning (eCP) algorithm [11], is an example of a vector quantifier that attempts to balance cluster sizes for improved performance, thus aiming to satisfy all three requirements; we conclude that eCP is a prime candidate for large-scale IL.

These high-dimensional indexes are global, making it difficult to explore or search subcollections, formed for instance by filters being applied to the retrieved candidates. For eCP, the clusters selected during retrieval may not have enough relevant items to present the user with. To avoid such scenarios, incremental retrieval using a priority queue is deployed to retrieve additional clusters [15], [39]. Incremental retrieval reuses information from previous searches to reduce the work of running a search from scratch [21]. It is utilised in path-finding algorithms such as A^* and LPA^* [21], [29]. Furthermore, for improving search in k-NN approaches [5] they have even been used for hierarchical high-dimensional indexes to improve search [31] and determining the best k values within a maximum distance [15], [39]. We conjecture that incorporating this into the retrieval process will be greatly beneficial when using indexes with restrictions on the search space.

III. The Exquisitor Approach

Figure 2 shows the proposed Exquisitor approach. The top (non-highlighted) elements represent a general IL process, the bottom (highlighted) elements represent the contributions of Exquisitor, where the trained classifier scores and selects the most relevant index cells from a high-dimensional index.

The index restricts search space to avoid processing the entire collection, introducing a trade-off between quality and latency. The user can also apply filters on metadata. In essence, the interactions of relevance judgements and applying filters between user and system form *dynamic subcollections* to

Algorithm 1 Priority Queue

```
Input: hyperplane $q$, clusters to return $b$, root of index $\theta$
Output: $b$ clusters based on farthest neighbors
procedure: SearchPQ($q$,$b$)
  $sel \leftarrow 0$ // Amount of clusters selected
  $res \leftarrow []$
  while $pq$ is not empty $\wedge$ $sel < b$ do
    $e \leftarrow$ pq.dequeue()
    if isLeaf($e$) then
      $res$.append($e$)
      $sel \leftarrow sel + 1$
    else
      foreach child $c$ of $e$ do $pq$.enqueue($c$, $dist$($q$,$c$))
    end if
  end while
  return $res$
procedure: SearchIndex($q$,$b$)
  $pq \leftarrow$ PQ($\theta$,$dist$($q$,$\theta$))
  return SearchPQ($q$,$b$)
```

search within, that may fall outside the index restriction. As filters are applied to the returned items of the index cells, the trade-off can result in zero quality if no items pass the filters. On such occasions, Exquisitor uses incremental retrieval along with query optimisation policies to expand the search space.

A. High-Dimensional Index

In the offline phase, the collection is prepared by first extracting visual semantic concepts using a deep convolutional neural network. This is followed by a compression that selects the top 7 features of an item and stores them in three 64-bit integers [49]. On this representation, a modified version of extended Cluster Pruning (eCP) is used to build an approximate high-dimensional index, which is capable to find k-farthest neighbors in the compressed space.

The extended Cluster Pruning algorithm builds the index by using the first step in k-means to form k clusters, where $k = N_{items}/ts$, with ts being the soft target size for the clusters (typically $ts = 100$). ts makes it possible to predict and control the latency when traversing the index [44]. Once cluster representatives are chosen, the clusters are filled. Then, a hierarchical tree with L levels is built top-down from the representatives, where the bottom-most level has $l_L = k$ clusters and the levels above have $l_{i-1} = \sqrt{l_i}$ [12], [11]. For multimodal data, an index is created per modality.

In the online phase, Exquisitor uses the user-provided labels to train its linear SVM classifier which is fast and reliable and does not require many training examples. The hyperplane from the SVM is then used to select clusters from the eCP index. Without applying incremental retrieval this is done by selecting the most relevant b node/clusters at each level in the hierarchy, similar to the breadth-first search. The b clusters are processed in segments, which mainly improve result quality when multiple modalities are used, preventing one modality taking over. After the clusters are processed, late fusion using rank aggregation is performed and the top items are presented to the user.

B. Incremental Retrieval

The user applies filters to the items returned from the b clusters, leading to a potential pitfall when the filters form a subcollection which is only partially, or not at all within the retrieved clusters. This is a general problem even for queries where many related items are found but the relevant ones are not within the b clusters. The b restriction, which brings significant latency control benefits, may thus be responsible for reduction or even elimination of relevant query results.

This is solved by incremental retrieval that gets additional b clusters when not enough items are found to return. We add a priority queue (*PQ*) to maintain the search state and provide means for incremental retrieval. When the initial b clusters do not have enough items, the search switches to using best-first-search which processes the most relevant nodes and leaves, regardless of their level in the index hierarchy. This is more search-oriented than the breadth-first-search.

A pseudo algorithm of the *PQ* implementation can be seen in Algorithm 1. The priority queue ranks the items based on the Euclidean distance of the cluster representative to the SVM decision boundary, where the largest distance in the positive direction is considered most relevant. Once the decision boundary is formed, the first element is inserted in the queue, which is the root of the index. The queue dequeues the most relevant element; if the element is a node it enqueues its children, and if it is a cluster it is added to the result list. Once b clusters are found, the interactive learning process continues. In the case where we do not have enough items to return, we only need to call SearchPQ(q,b) to get b more clusters.

As a measurable criterion to determine whether enough clusters are selected, we use $T = \frac{b \cdot ts}{2^{exp}}$, where $b \cdot ts$ is the expected number of items within the b clusters based on target size ts, and exp is the expansion iteration. This dynamic threshold reduces unnecessary expansions.

C. Query Optimisation Policies

We define query optimisation policies that derive an estimated count representing the total number of useful items for the b clusters. If the count is below threshold T, then b is doubled. Doubling b ensures that cases where the relevant subcollection is far away is found faster than with a constant b increase. This expansion continues until enough items pass the threshold or all the clusters have been processed.

Count Threshold (*CT*): The baseline estimated count (CT) simply accumulates the number of items from the b clusters.

Global Remaining Count (*GRC*): This policy tracks and updates the amount of items remaining in a cluster when any of its items are presented to the user. The policy is independent of filter knowledge as it only focuses on counting the number of items of the b selected clusters.

Filter Remaining Count (*FRC*): This policy monitors which filters are applied in the current round and how many items pass from the selected clusters. Two sets are used, one for tracking the exact amount of items passing each cluster and another for the number of items presented to the user from the cluster. These two sets are checked to get the exact number of remaining items passing the filters in the cluster. If a cluster has not been checked with the active filters then GRC is used for that cluster's count. The cases can be seen in Equation 1, where EC is the number of items passing the cluster and RI is the number of items returned from the cluster.

$$FRC(C_i) = \begin{cases} EC(C_i) - RI(C_i), & \text{if } EC(C_i) \text{ exists} \\ GRC(C_i), & \text{otherwise} \end{cases} \tag{1}$$

Estimated Remaining Count (*ERC*): The eCP index is typically built with a three level deep hierarchy, where the leaf nodes are the clusters. This query optimisation policy aims to calculate how many items exist within a selected cluster with a given filter combination. It uses a probability based on the number of items with the given combination and the total number of items from a set level in the index hierarchy. If the level from which the probabilities are calculated represents the leaf nodes then the count will be the exact number of items passing the filter combination. Since this policy will have an inherently high memory cost, it is better to choose levels above the leaf level to reduce this cost. Equation 2 shows how the count is derived, where C_i is the cluster, FC is the set of active filter combinations, $Pr(C_i, FC_i)$ represents the probability of how many items C_i is possibly containing with FC_i, and lastly $GRC(C_i)$ is the Global Remaining Count for C_i.

$$ERC(C_i) = \sum_{j=0}^{FC} Pr(C_i, FC_j) \cdot GRC(C_i) \tag{2}$$

All Remaining Count (*ARC*): The final policy is *All Remaining Count* which uses all described policies. Given a cluster C_i, a series of prioritised checks are performed. It starts by checking whether C_i has been fully seen, and in case it has not, it further checks whether or not to return FRC, GRC or ERC. This gives a more accurate estimated count.

IV. Experimental Setup

In this section we describe and motivate the conducted experiments that analyse the performance of Exquisitor with regards to analytical tasks, and then provide an overview of the baseline approaches we compare Exquisitor against.

A. Overview of Experiments

We report three experiments. The first experiment (Section V) focuses on classifier flexibility, evaluating the ability of Exquisitor to discover new concepts in data at scale. The second experiment (Section VI) focuses on the scalability of Exquisitor for exploration-oriented tasks in large-scale collections. In the third experiment (Section VII), we analyze the performance of Exquisitor for search-oriented tasks that dynamically form subcollections.

All three experiments use the ARC policy, however, note that for the first two experiments, which do not consider filters, this defaults to GRC. In Section VIII, we perform an ablation study of the query optimisation policies for the third experiment. For all experiments the artificial user judging the suggestions is presented with 25 items.

The first two experiments are exploration-oriented where the goal is to find as many relevant items as possible. As such, the key metric for the experiment is average precision per interaction round. The third experiment is search-oriented, where the task is to find the first relevant item within a subcollection. Thus, the key metric is the number of interaction rounds to complete the task. We also impose a limit on the number of interaction rounds, and in case tasks do not finish within this limit, average recall is also a metric of interest. For all experiments, average latency per interaction round is the most important performance metric. All collections are represented by 1,000 ILSVRC [38] concepts extracted with a convolutional neural network [7].

B. Baseline Approaches

In the experiments, we compare the Exquisitor approach to the following approaches:

Blackthorn: To the best of our knowledge, Blackthorn [49] is the only direct competitor on interactive learning at 100M scale. Blackthorn uses no indexing or prior knowledge about the structure of the collection, deploying instead data compression and multi-core processing for achieving scalability.

kNN+eCP: This baseline is representative of pure query-based approaches using a k-NN query vector to search the index, based on relevance weights [35], [25], an approach that was initially introduced for text retrieval [36] but has been adapted for CBIR with relevance feedback [37].

SVM+LSH, kNN+LSH: These baselines represent SVM-based and k-NN-based approaches using LSH indexing. We replace the eCP index with a multi-probing LSH index [30]. The LSH index has many parameters for tuning performance, such as L number of hash tables, B number of buckets in each table, and p number of buckets to read from each table at query time. For a fair comparison and best performance, we have set the parameters as in eCP.

V. Exp. 1: Discovering ImageNet Concepts

We evaluate IL flexibility using a zero-shot-inspired protocol for interactive learning on the popular ImageNet dataset, which consists of 14,198,361 images categorized into approximately 21,000 WordNet synsets (synonym sets) [8].

A. Experimental Protocol

Zero-shot learning is a method which trains a classifier to find target classes either without including them in the training set or by including them in the training set, but without labels. We arbitrarily select 50 concepts and create an artificial user (henceforth called actor) for each. The objective of each actor is to find images belonging to their associated concept. To clearly capture the effects of knowing the concept versus not knowing it, this experiment is run once with the value of the concept represented in the collection, acting as a baseline, and once with the concept being unknown.

The experiment has a total of 50 runs per actor, each run with 10 interaction rounds starting from a different random subset. Each run is initialized by 10 random positive images and 100 random negative images, simulating an ongoing interactive session.

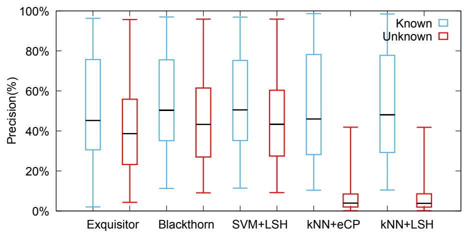


Fig. 3: Average precision per round across all ImageNet actors for each interactive learning approach. The blue boxes depict the known case, while the red depict the unknown case.

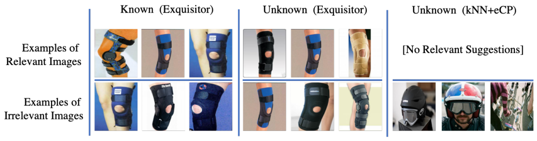

Known (Exquisitor)
Unknown (Exquisitor)
Unknown (kNN+eCP)
Examples of Relevant Images
Examples of Irrelevant Images
[No Relevant Suggestions]

Fig. 4: Examples of relevant and irrelevant suggestions for different approaches from an ImageNet actor with the "knee pad" concept.

B. Results

Figure 3 shows the results from this experiment, comparing Exquisitor against all baseline approaches. The blue boxes show the average precision distribution for each actor in the case where the system knows the concepts, while the red boxes show the precision distribution for the case where the concepts are unknown. As expected, the results where the concepts are unknown, are worse than when they are known. A noticeable trend, however, is that all approaches using the linear SVM only fall by about 10% whereas the k-NN approaches largely fail to recover the semantic concepts if unknown. As it is likely that items with the removed concept have some other related concepts, the linear SVM is able to capture this, while the k-NN approach does not adapt as well.

Note that since many images are categorized into multiple WordNet synsets, the ImageNet collection contains duplicate items that are labelled differently, which can lead to false negatives. This is clearly evident by Figure 4 which shows suggestions from an interaction round where the top row are relevant items and the bottom row irrelevant items. These are from a run for Exquisitor (known/unknown) and kNN+eCP (unknown). In the unknown case Exquisitor, using the linear SVM, finds false negatives, while the k-NN finds no relevant suggestions and the irrelevant items are true negatives.

In terms of average latency, all approaches using high-dimensional indexes reach around 8 ms with 1 CPU core, over-performing Blackthorn that requires 17 ms. This highlights the advantages of an index-based approach.

VI. Exp. 2: Performance at YFCC100M Scale

This experiment uses an experimental protocol from the literature, defined over the YFCC100M collection [49], to illustrate the interactive performance and retrieval quality of Exquisitor at very large scale. YFCC100M consists of 99,206,564 million images along with their associated annotations (i.e. title, tags and description), and a range of metadata, including geo-location and timestamps. The visual content is again represented by the 1,000 ILSVRC concepts. The textual content is encoded by a) treating the title, tags, and description as a single text document, and b) extracting 100 LDA topics for each image.

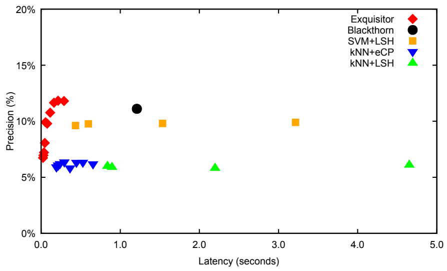


Fig. 5: Average precision vs. latency over 10 rounds of analysis across all YFCC100M actors. Exquisitor, kNN+eCP: $b = 1 - 512$. LSH: $L = 10$, $B = [2^{10}, 2^{18}]$, $p = [15, 40]$.

A. Experimental Protocol

This protocol is inspired by the MediaEval Placing Task [26], [6], in which actors look for images from one of 50 world cities. Regarding evaluation metrics, it is worth noting that due to both the scale of YFCC100M and its unstructured nature, precision is lower in absolute terms than in experiments involving small and well-curated collections. The runs are performed consistent with the experimental protocol presented in Section V and involve 10 interaction rounds with an initial positive and negative set.

For the eCP index, b ranges from 1 to 512, with increments by power of 2. For LSH, we experiment with a variety of parameter settings, including those aimed at reflecting a similar cell size distribution as eCP.

B. Results

Figure 5 shows the results from this experiment. Exquisitor achieves the best performance on both precision and latency. The closest approach in terms of precision is Blackthorn. However, Blackthorn has much higher latency. The improvement in precision between Exquisitor and Blackthorn is due to the selection and processing of clusters in Exquisitor, making it more consistent than Blackthorn, which scans the entire collection. SVM+LSH approaches the performance of Exquisitor in terms of latency when $b = 512$, but has significantly lower precision. We conjecture that there may be a parameter setting that allows the SVM+LSH approach to also get better precision but there is no guarantee that such a setting will not overfit to the experiment. As for the k-NN approaches, both eCP and LSH reach the same precision at roughly 6% but again fall into the issue of focusing on the wrong concepts and failing to improve quality, regardless of index settings.

TABLE I: Example LSC and VBS tasks with expected results

Task	Description	Result
LSC25	Find the time when I was looking at an old clock, with flowers visible. There was a lamp also, and a small blue monster (perhaps a long rabbit) watching me. Maybe there were two monsters. It was a Monday or a Thursday. I was at home and in a bedroom.	
TKIS23 (VBS)	Red elevator doors opening, a bike leans inside, doors closing and reopening, bike is gone. Zoom-in on bike, zoom-out from empty elevator. The bike is silver, the text 'ATOMZ' is visible.	

VII. Exp. 3: Dynamic Subcollections

This experiment focuses on the performance of Exquisitor for tasks that lead to dynamic subcollections. For this experiment, we use two small collections, LSC2019 [13] and VBS2020 [40], LSC and VBS in short, that contain interactive tasks with query descriptions that involve dynamic subcollection exploration, when users apply filters on metadata. To test the ability of Exquisitor when exploring such subcollections at scale, we merge both collections into YFCC100M, forming the LSC+YFCC and VBS+YFCC collections. An example task for each collection can be seen in Table I.

LSC2019 is a collection from the Lifelog Search Challenge 2019, featuring 41,666 images [13] along with metadata such as location, day and time. In total, it has 24 interactive tasks with corresponding ground truths, where each task aims to find one relevant image matching a textual query describing events from the lifelogger.

VBS2020 is a collection from the Video Browser Showdown 2020, consisting of 1000 hours of video from Vimeo, which are segmented and represented by 1,082,567 keyframes. The metadata of VBS2020 entails video level categories and tags defined by the user uploading the video. The categories are selected from a fixed set, while tags are created with no restrictions. A keyframe-level metadata for the number of faces detected is also extracted [40]. There are 13 interactive tasks with corresponding ground truths for this collection.

A. Experimental Protocol

For these experiments we deploy actors on the protocol of the real LSC and VBS challenges. The actors are equipped with specific labeling and filtering strategies corresponding to real users with various levels of collection knowledge and URF expertise [20]. The levels are: *Novice*, a new user interacting with the collection; *Expert*, a user who has worked with the system and collection for a long time; and *Data Author*, a user that was part of creating or curating the collection.

Labeling strategies determine how many positive and negative examples are chosen in each round, and whether items can be replaced from the positive/negative sets if better examples are found. We use the ±*AccRep* strategy for LSC and +*FixRep-AccAdd* for VBS, as these were shown to provide the best result quality [20]. With ±*AccRep*, the actor *accumulates* p positive examples and n negative examples in each interaction round, and can *replace* items if better examples are found. +*FixRep-AccAdd* uses a *fixed-size* positive set of p items, requiring the actor to only replace items once it is full, while the negative set accumulates as before, but the actor can only *add* n negative examples. In all experiments, we use the recommended settings of $p = 5$ and $n = 15$ [20].

As the intention behind these experiments is to form dynamic subcollections we use the *Expert* filtering strategy, which resembles a user with enough domain knowledge to infer filters not necessarily present in the query text and avoids applying wrong filters. Note that there are 5 tasks in LSC with the *Expert* filtering strategy that do not set any filters, and thus do not form dynamic subcollections. We omit these 5 tasks from the experiments.

The experimental protocol simulates users that deal with tasks from the start of the interactive learning process. To better reflect the real-world time constraints, the number of rounds is limited to 500 rounds in the protocol. This may be considered a generous number, but a high number allows better understanding of the effects of the relevance feedback process in dynamic subcollections. Each task is run 50 times with different starting points.

In this experiment we compare Exquisitor with a kNN+eCP approach, which has also been configured with incremental retrieval and query optimisation policies. The b parameter for this experiment is set to 256. In addition, Blackthorn is run on the smaller LSC and VBS collection only, as the previous experiment has shown that it requires more computing resources at this scale and still takes longer with similar to less quality. We exclude experiments with LSH as the previous experiment has shown that it is inferior to Exquisitor, but also due to a need to set multiple parameters which can be collection specific making it far less flexible than the other approaches.

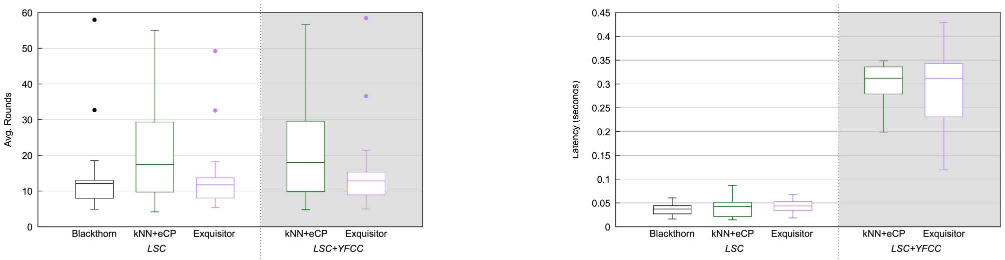
B. Results

LSC and LSC+YFCC: Figure 6(a) shows the distribution of average rounds to complete tasks in the LSC2019 benchmark, using *Expert* filters guaranteeing dynamic subcollections.

Consider first the left side of the figure, which focuses on the small LSC collection. Blackthorn and Exquisitor have the best performance, with the majority of tasks taking 5-20 rounds of interaction, while kNN+eCP performs significantly worse. Note that, since all approaches solve all tasks in fewer than 60 rounds, recall is 1.0 in all cases.

The right side of Figure 6 shows the average rounds to complete tasks within LSC+YFCC, which is more than 1000x larger than LSC. The performance of both kNN+eCP and Exquisitor is only affected very modestly in the larger collection, and as before Exquisitor's performance is nearly identical to that of Blackthorn on the smaller collection. Also note that Exquisitor manages to get better results for the longest task in the case of the smaller LSC collection, which is due to the b restriction limiting items. Considering the collection size difference, we believe that the nearly identical performance for Exquisitor is a remarkable result.

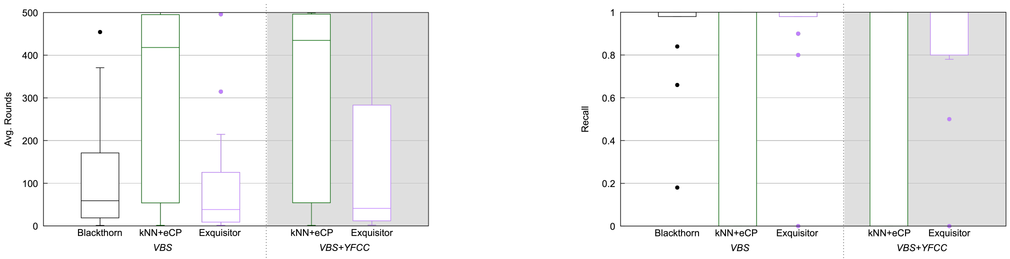
Figure 6(b) shows the distribution of average response time per round for the tasks. The collection size determines the



(a) Average rounds to complete tasks

(b) Average latency per round

Fig. 6: Performance comparison between Exquisitor and kNN+eCP with $b = 256$ on the LSC and LSC+YFCC collection, against Blackthorn on the LSC collection (black).



(a) Average rounds to complete tasks

(b) Average recall

Fig. 7: Performance comparison between Exquisitor and kNN+eCP with $b = 256$ on the VBS and VBS+YFCC collection, against Blackthorn on the VBS collection (black).

processing time and kNN+eCP and Exquisitor have similar latencies, since they both use the same index. In all cases, the average latency is below 0.5 seconds, and with an average of 0.29 seconds across all runs.

VBS and VBS+YFCC: Figure 7(a) shows the average rounds to complete tasks for the VBS2020 benchmark. The left side again focuses on the smaller collection of VBS. Here we see that Blackthorn and Exquisitor are able to complete tasks more consistently than kNN+eCP with Exquisitor being best overall.

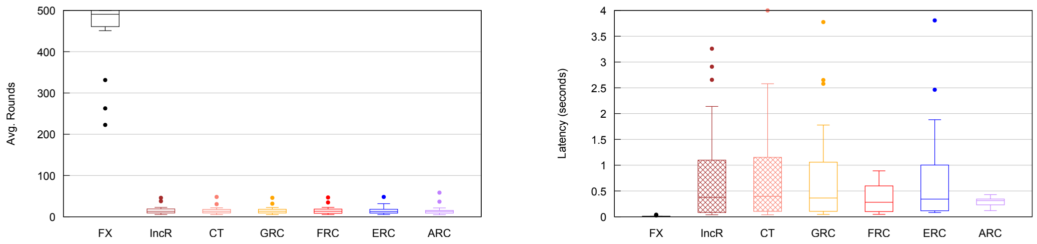
The right side focuses on the larger collection VBS+YFCC. kNN+eCP's performance is unchanged but still bad. Exquisitor performs better on average but exhibits a performance drop in comparison to Blackthorn. This is partly due to the different distribution of items in the index for the larger collection and because of positives and negatives from the YFCC collection in the first few rounds. It is still far better than kNN+eCP. Finally we observe that the latency for each approach (not shown) is similar to the LSC and LSC+YFCC respectively.

As the VBS collection has less carefully curated metadata than LSC, there are situations where the item(s) of interest are not found within 500 rounds. Therefore it is important to analyse the recall for this benchmark. Figure 7(b) depicts average recall per run, where the left side shows the result for the smaller collection, and the right side for the larger. Focusing first on the smaller, we see that even when the entire collection is available, using Blackthorn, there are outlier tasks that only complete as little as 19% of the runs. kNN+eCP performs significantly worse. Exquisitor's performance is similar to Blackthorn for the smaller collection, with one of the outlier tasks not completing and the other two outlier tasks having better recall than Blackthorn. As for the VBS+YFCC, Exquisitor again has one task that is not completing, while the rest have a fairly high recall. The impact on recall is due to the query optimisation policy ARC expanding too much and returning noise which distracts the classifier.

VIII. Ablation Study

We perform an ablation study to investigate the impact of incremental retrieval and different query optimisation policies in Exquisitor, using the experimental protocol from Experiment 3. The baseline, fixed (*FX*) version of Exquisitor has both incremental retrieval and query optimization policies disabled. We then consider versions where incremental retrieval is enabled with different query optimisation policies.

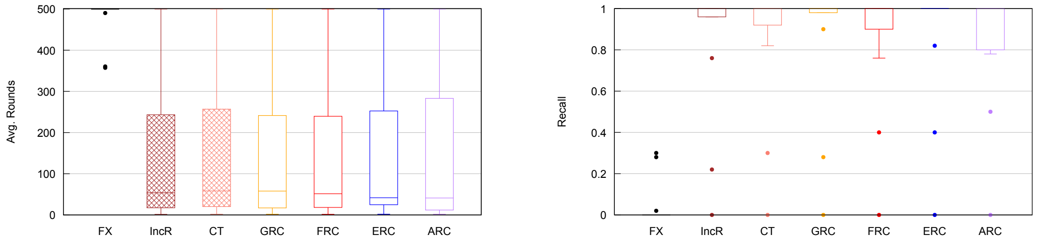
LSC+YFCC: Figure 8(a) shows the average rounds to complete tasks within LSC+YFCC. The *FX* version (the leftmost box), fails to complete majority of the tasks, and even those that do complete require on average more than 200 rounds. In comparison, Exquisitor with *IncR* alone, or with query optimisation policies is consistently far below that. For this experimental protocol there is no significant difference in rounds to complete tasks for any combination of *IncR* and query optimisation policy. However, there are differences in latency, shown in Figure 8(b), which depicts the average



(a) Average rounds to complete tasks

(b) Average time per round in seconds

Fig. 8: Performance of Exquisitor with various settings on the LSC+YFCC collection using *Expert* filters.



(a) Average rounds to complete tasks

(b) Average recall per run

Fig. 9: Performance of Exquisitor with various settings on the VBS+YFCC collection using *Expert* filters.

latency per round. *FX* is the fastest, which is not surprising as no additional clusters are selected beyond b. *IncR* raises the average latency to above 1 second for majority of the tasks, as it only gets b additional clusters every increment. For the query optimisation policies, CT and GRC are similar to *IncR*, while FRC is slightly better and below 1 second. ARC is the most consistent policy and has an average latency of 0.29 seconds which is the best of all the policies. With regards to memory cost of policies for LSC+YFCC, GRC uses 8 MB, FRC uses 16 MB and ERC uses 11 MB. With such a low cost, memory has no real impact on the choice of policy.

VBS+YFCC: Figure 9(a) shows the average rounds to complete the tasks within VBS+YFCC. As mentioned in Experiment 3, the VBS+YFCC protocol does not have strong filters as the LSC+YFCC protocol, leading to more rounds in average. We see a similar performance impact between *FX* and *IncR*, where the former is incapable of completing almost any task, the latter manages to complete all but one. The query optimisation policies results are again quite similar, with GRC being the best, along with ERC. ARC does worse in terms of average rounds to complete tasks, however it is still the fastest with an average latency of 0.25 seconds per round. Figure 9(b) shows the distribution of average recall per run for tasks in the protocol. It is again evident that the best policy for this task is GRC along with ERC. The reason for ERC being better than ARC here is due to the FRC policy's influence resulting in additional expansions that lead to noise. The deciding factor for the best policy for VBS+YFCC, comes down to latency or memory usage. The memory usage for ERC in VBS+YFCC is significantly different than LSC+YFCC. While GRC and FRC use the same memory, ERC uses 2.3 GB. This increase is due to VBS having far more options for each filter, which in turn requires more space for statistics.

Summary: For any collection the base policy is GRC. For well curated collections ARC can be used, but as the memory cost of ERC is tied to the amount of items and existing metadata filter combinations, ARC can potentially have a high memory cost. Therefore, FRC is the better choice for well curated collections with many filter combinations.

IX. Conclusion

In this paper, we have introduced Exquisitor, a responsive, accurate, flexible and scalable interactive learning approach, that manages to bridge the semantic, pragmatic and scale gap between human and machine. Exquisitor outperforms the state of the art on precision, recall, and latency, with low latency maintained even on very large collections. At the same time, Exquisitor is computationally efficient, using 16x less resources than its direct state-of-the-art competitor, Blackthorn. Finally, since our experimental protocols included a variety of collections and tasks, we have demonstrated that Exquisitor is versatile. We believe that this establishes Exquisitor as the new state of the art and unlocks potential for truly interactive search and exploration applications at large scale.

References

[1] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Proc. of the IEEE Symp. on the Foundations of Comp. Science*. IEEE, 2006, pp. 459–468.

[2] A. Babenko and V. S. Lempitsky, "The inverted multi-index," *IEEE TPAMI*, vol. 37, no. 6, pp. 1247–1260, 2015.

[3] A. Barbu, D. Mayo, J. Alverio, W. Luo, C. Wang, D. Gutfreund, J. Tenenbaum, and B. Katz, "Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models," in *Proc. NIPS*, 2019.

[4] C. Böhm, S. Berchtold, and D. A. Keim, "Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases," *ACM Comput. Surv.*, vol. 33, no. 3, p. 322–373, Sep 2001.

[5] B. Bustos and G. Navarro, "Improving the space cost of k-nn search in metric spaces by using distance estimators," *Multim. Tools Appl.*, vol. 41, no. 2, p. 215–233, Jan. 2009.

[6] J. Choi, C. Hauff, O. V. Laere, and B. Thomee, "The placing task at MediaEval 2015," in *Proc. MediaEval 2015 Workshop*. CEUR, 2015.

[7] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. CVPR*. IEEE, 2017, pp. 1800–1807.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, 2009, pp. 248–255.

[9] L. Gao, J. Song, X. Liu, J. Shao, J. Liu, and J. Shao, "Learning in high-dimensional multimedia data: the state of the art," *Multim. Sys.*, vol. 23, no. 3, pp. 303–313, 2017.

[10] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proc. VLDB*, 1999, pp. 518–529.

[11] G. Þ. Gudmundsson, L. Amsaleg, and B. Þ. Jónsson, "Impact of storage technology on the efficiency of cluster-based high-dimensional index creation," in *Proc. International Conference on Database Systems for Advanced Applications (DASFAA)*. Springer, 2012, pp. 53–64.

[12] G. Þ. Gudmundsson, B. Þ. Jónsson, and L. Amsaleg, "A large-scale performance study of cluster-based high-dimensional indexing," in *Proc. of Int. Workshop on Very-large-scale Multim. Corpus, Mining and Ret.* ACM, 2010, pp. 31–36.

[13] C. Gurrin, K. Schoeffmann, H. Joho, A. Leibetseder, L. Zhou, A. Duane, D. Nguyen, D. Tien, M. Riegler, L. Piras *et al.*, "Comparing approaches to interactive lifelog search at the lifelog search challenge (lsc2018)," *ITE Trans. on Media Tech. and App.*, vol. 7, no. 2, pp. 46–59, 2019.

[14] X. He, O. King, W.-Y. Ma, M. Li, and H.-J. Zhang, "Learning a semantic space from user's relevance feedback for image retrieval," *IEEE Trans. on Cir. and Sys. for Vid. Tech.*, vol. 13, no. 1, pp. 39–48, 2003.

[15] G. R. Hjaltason and H. Samet, "Incremental distance join algorithms for spatial databases," in *Proc. ACM SIGMOD*, 1998, pp. 237–248.

[16] T. Huang, C. Dagli, S. Rajaram, E. Chang, M. Mandel, G. E. Poliner, and D. Ellis, "Active learning for interactive multimedia retrieval," *Proc. IEEE*, vol. 96, no. 4, pp. 648–667, 2008.

[17] M. Huijser and J. C. v. Gemert, "Active decision boundary annotation with deep generative models," in *IEEE ICCV*, 2017, pp. 5296–5305.

[18] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE TPAMI*, vol. 33, no. 1, pp. 117–128, 2011.

[19] B. Þ. Jónsson, M. Worring, J. Zahálka, S. Rudinac, and L. Amsaleg, "Ten research questions for scalable multimedia analytics," in *Proc. MMM*. Springer, 2016, pp. 290–302.

[20] O. S. Khan, B. Þ. Jónsson, J. Zahálka, S. Rudinac, and M. Worring, "Impact of interaction strategies on user relevance feedback," in *Proc. ICMR*. ACM, 2021, p. 590–598.

[21] S. Koenig, M. Likhachev, Y. Liu, and D. Furcy, "Incremental heuristic search in ai," *AI Magazine*, vol. 25, no. 2, pp. 99–99, 2004.

[22] C. Kofler, M. Larson, and A. Hanjalic, "User intent in multimedia search: A survey of the state of the art and future challenges," *ACM Computing Surv.*, vol. 49, no. 2, Aug 2016.

[23] A. Kovashka, D. Parikh, and K. Grauman, "Whittlesearch: Interactive image search with relative attribute feedback," *IJCV*, vol. 115, no. 2, pp. 185–210, Nov 2015.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Neural Information Processing Systems*. Curran Associates Inc., 2012, p. 1097–1105.

[25] M. Lan, C. L. Tan, J. Su, and Y. Lu, "Supervised and traditional term weighting methods for automatic text categorization," *IEEE TPAMI*, vol. 31, no. 4, pp. 721–735, 2008.

[26] M. Larson, M. Soleymani, P. Serdyukov, S. Rudinac, C. Wartena, V. Murdock, G. Friedland, R. Ordelman, and G. J. F. Jones, "Automatic tagging and geotagging in video collections and communities," in *Proc. ICMR*. ACM, 2011, pp. 51:1–51:8.

[27] D. Li and E. Kanoulas, "When to stop reviewing in technology-assisted reviews: Sampling from an adaptive distribution to estimate residual relevant documents," *ACM Trans. Inf. Syst.*, vol. 38, no. 4, Sep 2020.

[28] Z. Liu, J. Wang, S. Gong, H. Lu, and D. Tao, "Deep reinforcement active learning for human-in-the-loop person re-identification," in *IEEE ICCV*, 2019.

[29] Y. Lu, X. Huo, O. Arslan, and P. Tsiotras, "Incremental multi-scale search algorithm for dynamic path planning with low worst-case complexity," *IEEE Trans. on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 6, pp. 1556–1570, 2011.

[30] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li, "Multi-probe LSH: efficient indexing for high-dimensional similarity search," in *Proc. VLDB*, 2007, pp. 950–961.

[31] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE TPAMI*, vol. 36, no. 11, pp. 2227–2240, 2014.

[32] L. Paulevé, H. Jégou, and L. Amsaleg, "Locality sensitive hashing: A comparison of hash function types and querying mechanisms," *Pattern Recognition Letters*, vol. 31, no. 11, pp. 1348–1358, 2010.

[33] L. Putzu, L. Piras, and G. Giacinto, "Convolutional neural networks for relevance feedback in content based image retrieval," *Multim. Tools and App.*, vol. 79, no. 37, pp. 26 995–27 021, 2020.

[34] W. Ribarsky and B. Fisher, "The human-computer system: Towards an operational model for problem solving," in *Proc. Hawaii Int. Conf. on Sys. Sciences*, 2016, pp. 1446–1455.

[35] S. E. Robertson and K. Spärck Jones, "Simple, proven approaches to text retrieval," Univ. of Cambridge, Comp. Lab., Tech. Rep., 1994.

[36] J. J. Rocchio, "Relevance feedback in information retrieval," University of Harvard, Computer Laboratory, Tech. Rep., 1965.

[37] Y. Rui, T. S. Huang, and S. Mehrotra, "Content-based image retrieval with relevance feedback in MARS," in *Proc. of Int. Conf. on Image Processing*, vol. 2. IEEE, 1997, pp. 815–818.

[38] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int Journal of Comp. Vis.*, vol. 115, no. 3, pp. 211–252, Dec 2015.

[39] H. Samet, "K-nearest neighbor finding using maxnearestdist," *IEEE TPAMI*, vol. 30, no. 2, pp. 243–252, 2008.

[40] K. Schoeffmann, "A user-centric media retrieval competition: The Video Browser Showdown 2012-2014," *IEEE MM*, vol. 21, no. 4, pp. 8–13, 2014.

[41] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE TPAMI*, vol. 22, no. 12, pp. 1349–1380, 2000.

[42] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[43] Y. Tao, K. Yi, C. Sheng, and P. Kalnis, "Quality and efficiency in high dimensional nearest neighbor search," in *Proc. ACM SIGMOD*, 2009, pp. 563–576.

[44] R. Tavenard, H. Jégou, and L. Amsaleg, "Balancing clusters to reduce response time variability in large scale image search," in *Int. Workshop on Content-Based Multim. Indexing*. IEEE, 2011.

[45] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, "YFCC100M: The new data in multimedia research," *Comm. ACM*, vol. 59, no. 2, p. 64–73, Jan. 2016.

[46] R. Tronci, G. Murgia, M. Pili, L. Piras, and G. Giacinto, *ImageHunter: A Novel Tool for Relevance Feedback in Content Based Image Retrieval*. Springer Berlin Heidelberg, 2013, pp. 53–70.

[47] Y. Yang, Z. Ma, F. Nie, X. Chang, and A. G. Hauptmann, "Multi-class active learning by uncertainty sampling with diversity maximization," *IJCV*, vol. 113, no. 2, pp. 113–127, Jun 2015.

[48] D. Yoo and I. S. Kweon, "Learning loss for active learning," in *Proc. CVPR*, 2019.

[49] J. Zahálka, S. Rudinac, B. Þ. Jónsson, D. C. Koelma, and M. Worring, "Blackthorn: Large-scale interactive multimodal learning," *IEEE TMM*, vol. 20, no. 3, pp. 687–698, 2018.

[50] J. Zahálka and M. Worring, "Towards interactive, intelligent, and integrated multimedia analytics," in *Visual Analytics Science and Technology (VAST)*. IEEE, 2014, pp. 3–12.

[51] J. Zahálka, M. Worring, and J. J. van Wijk, "II-20: Intelligent and pragmatic analytic categorization of image collections," 2020.

[52] L. Zhu, Z. Huang, Z. Li, L. Xie, and H. T. Shen, "Exploring auxiliary context: discrete semantic transfer hashing for scalable image retrieval," *IEEE Trans. on Neural Networks and Learning Sys.*, vol. 29, no. 11, pp. 5264–5276, 2018.

Influence of Late Fusion of High-Level Features on User Relevance Feedback for Videos

Omar Shahbaz Khan
IT University of Copenhagen
Copenhagen, Denmark
omsh@itu.dk

Jan Zahálka
Czech Technical University in Prague
Prague, Czech Republic
jan.zahalka@cvut.cz

Björn Þór Jónsson[*]
Reykjavik University
Reykjavík, Iceland
bjorn@ru.is

ABSTRACT

Content based media retrieval relies on the multi-modal data representations. For videos, these representations focus on representations for the textual, visual, and audio modalities. While the modality representations can be used individually, combining their information can improve the overall retrieval experience. For video collections, retrieval focuses on either finding a full length video or specific segment(s) from one or more videos. For the former, the textual metadata along with broad descriptions of the contents are useful. For the latter, visual and audio modality representations are preferable as they represent the contents of specific segments in videos. To solve exploration and search tasks in larger collections, an interactive learning approaches such as user relevance feedback has shown promising results. When combining modality representations in user relevance feedback, often a form of late modality fusion method is applied. While this generally tends to improve retrieval, its performance for video collections with multiple modality representations of high-level features, is not well known. In this study we analyse the effects of late fusion using high-level features, such as semantic concepts, actions, scenes, and audio. From our experiments on three video datasets, V3C1, Charades, and VGG-Sound, we show that fusion works well, but depending on the task or dataset, excluding one or more modalities can improve results. When it is clear that a modality is better for a task, setting a preference to enhance that modality's influence in the fusion process can also be greatly beneficial. Furthermore, we show that mixing fusion results and results from individual modalities can be better than only performing fusion.

KEYWORDS

User Relevance Feedback, Multimedia Retrieval, Modality Representations, Late Fusion

ACM Reference Format:
Omar Shahbaz Khan, Jan Zahálka, and Björn Þór Jónsson. 2022. Influence of Late Fusion of High-Level Features on User Relevance Feedback for Videos. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

There has been significant growth in video collections over the past decade, primarily spearheaded by social media platforms such as YouTube, Facebook, and Instagram. More recently, TikTok and Shorts (YouTube) are having an even greater impact, as their format focuses on short videos, mostly below 30 seconds, leading to more frequent uploads. To search for a video, multiple search strategies can be used. For finding entire videos, textual search is often used, which relies on the available textual metadata. If a user is interested in finding videos containing an exact event or series of events, a more appropriate interactive retrieval strategy is warranted, that relies on data describing the contents within segments of videos. With massive collections it is unlikely that any user knows everything within it, which is why search strategies are not always the go-to interaction, but rather strategies that allow a mix of exploration and search. User relevance feedback is an interactive learning approach where a human user and machine work together to solve complex analytical tasks involving information needs exceeding the machine's understanding of the content, and information needs that shift between exploration and search [34]. The user specifies relevant and irrelevant items on a suggestion set obtained from the machine's current classifier. The feedback is used to update the interactive classifier that is then used to procure new potentially relevant suggestions. User relevance feedback is an approach that leans more towards exploration initially, but depending on the feedback can either slowly or quickly hone in on certain areas of the collection.

The information extracted from videos is inherently multimodal and can be divided into textual (metadata relating to the entire video, such as uploading user, description, categories, and tags), visual (the contents from shots within the original video, such as semantic concepts, actions, scenes, colors, and number of objects), and audio (such as the sound of music or actions occurring in shots). All modalities are potentially useful for building classification models, for performing similarity search or keyword search, and for serving as filters to narrow the scope of the collection. As the textual metadata is generally video-level information, it can negatively impact the retrieval by shifting it towards a specific video rather than the desired shots. Thus, representations that focus on shot-level details are better in the case of finding specific content occurring within videos [16, 20].

The interplay between modalities is often discussed with regards to umbrella terms, such as textual and visual. Within each of these modalities, however, multiple types of features can be extracted that represent different aspects of the content. For instance, the visual modality contains semantic concepts and actions, both of which can be used to describe a shot, that represent two completely

[*] Work done while the author was with the IT University of Copenhagen.

Conference'17, July 2017, Washington, DC, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

different notions. Therefore, treating them as individual modality representations during the retrieval process can be beneficial.

A common method for enhancing content-based retrieval is to combine the information between modalities. This can take the form of early fusion with a joint representation, either by concatenating the various representations into one [28], or by learning a new representation through deep learning, such as text-image embeddings [24] or audio-visual embeddings [6, 22]. Having a single representation for a multimedia item does simplify retrieval, in terms of storage but adds to computational complexity as more attributes are involved. Additionally, there is no easy way to switch a modality off, and the deep learning joint embeddings do require more time to train with a large annotation set.

Another form of fusion is late fusion, which fuses modalities during the retrieval phase or at the end of it, by either training a classifier using the different modalities as input, or by merging the results of each modality [8, 19, 23, 28]. The basic approach is to get the result set of each classifier and merge them using either the sum or product of the classifier scores [1, 17], or rank aggregation where the positions of items within each result set determines the place in the final set [10, 18, 32]. A major benefit of late fusion is that it allows control over modalities during retrieval, so if a representation is discovered to be favorable or unfavorable for a task, its influence can be increased or reduced accordingly. This can either be done on-the-fly by a user or by learning from the supplied query examples [23, 31]. As it is often the case that properties of all user tasks cannot be determined beforehand, the flexibility of late fusion can be valuable for an interactive approach.

The majority of the approaches in the literature focus on fusion across modalities [10, 15]. There are some that consider fusing multiple representations from a single modality but these have primarily focused on images and the visual modality [2, 3]. As the task of users can vary between exploration and search, user relevance feedback is an appropriate interactive approach to use for evaluating the effects of late fusion and the influence of modality representations accordingly. In large-scale user relevance feedback approaches, late fusion through rank aggregation has been shown to perform well for image collections exceeding 100 million items [18].

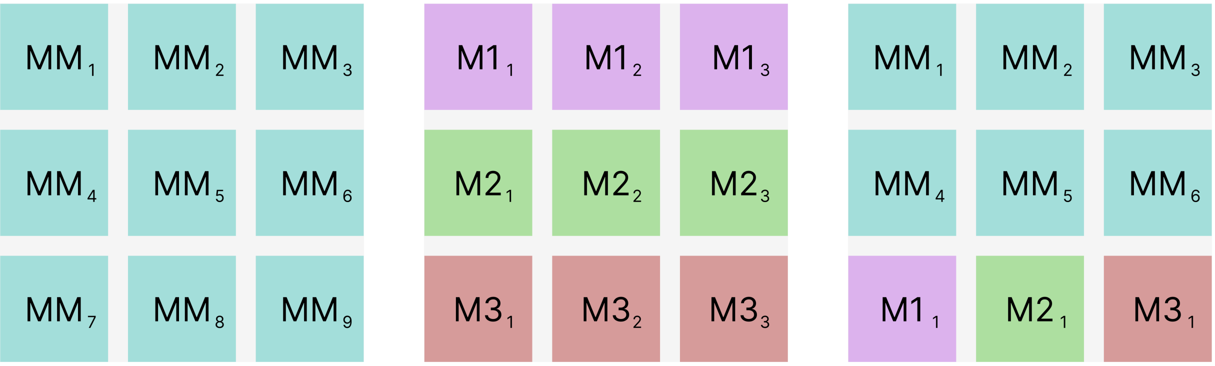
In this work, we investigate the effects of late fusion when using high-level features within and across modalities in user relevance feedback. We examine the current rank aggregation method and explore the effects of using fixed weights to amplify specific modalities. Additionally, we check whether not fusing the results has merit, by splitting the suggestion set into segments for each modality. Lastly, we consider a partial fusion method that mixes the suggestion set by having a fixed number of fused and non-fused slots. We conduct experiments on three video datasets, V3C1 [4], Charades [27], and VGG-Sound [6], focusing on tasks revolving around ad-hoc video search. Through our findings we show that the current rank aggregation method works well overall. However, when there is disparity in strength between modalities for a task, assigning more weight to the strongest modality improves quality. We also discover that mixing the suggestions with fused and non-fused items performs either on par with, or better than, the rank aggregation.

2 LATE FUSION METHODS

In retrieval it is common to return the top k items of a collection instead of all items, as it reduces computation and latency, and does not overwhelm the user. If only a single representation is used for the items in a collection, then the retrieval process simply needs to rank the items based on this and return the top k. With multiple modality representations of a data item it is more complicated, as they lead to different top k items, thus a late fusion step is required. Late fusion can be performed in many ways, ranging from straightforward approaches, that minimise computation and reduce latency, to approaches that perform elaborate computations based on details of the available information. The former is warranted in an interactive learning approach, since the latter results in increased response times which in turn lead to users losing focus [26]. This section outlines the fusion methods for interactive learning considered in this study.

A simple approach is to return only items that are present in two or more representations' top items, however, this may lead to fewer than k items being returned. Another approach is to make a union across all the items from each representation into a combined list R, where the top k items can be determined based on a function that accounts for all representations. If the query scores/distances of representations are of the same nature, then a possibility is to add the scores together and re-rank R based on this. However, it is unlikely that all representations hold this property, therefore, a better approach is **rank aggregation**. Rank aggregation is an approach where re-ranking is done for each representation and an aggregate score based on an items rank is calculated. This aggregate score is then used to re-rank R a final time, and return the top k. There are multiple ways to determine the aggregate score such as using the sum, average, median, and linear combination [11, 25]. In this study we use the sum of an item's ranks, as it requires less computation and shows the true influence of a representation.

The benefit of rank aggregation is that it treats each modality equally and is independent of different representations. However, treating each modality equally is not always a good choice, as different tasks may favor one modality over another. In such cases, some interactive approaches are able to disable bad modalities or use fixed weights to modify the impact of the modalities [9]. The weights may also be adjusted automatically based on the interactions with the user [30, 31], but such approaches are best when there is a clear indication of good examples belonging to a specific modality. In our case, the user's feedback is gathered from the multi-modal output of the top k items from R, as seen in Figure 1a, making it less transparent as to which modality strongly influenced them to be highly ranked. Setting fixed weights for each modality representation is possible but not realistic, as a regular user will have tremendous difficulties in determining them. Instead, to study the impact of weights we use a **weighted rank aggregation** approach where a user sets a preferred modality. This is a more realistic approach as the user only needs to know what a modality represents, such as concepts or actions, which is enough for them to determine whether a task favors one or the other. Essentially, this means that initially all modalities have a weight of w and if a preferred modality is set, its weight will be larger than the rest. When performing rank



(a) Rank Aggregation

(b) No Fusion with 3 modalities

(c) Partial Fusion with 3 modalities

Figure 1: Output from the different late fusion methods

aggregation, the modality weights are applied during the aggregate score calculation.

Some retrieval systems separate modalities entirely and allow the user to query specific modalities. If the user queries multiple modalities, then either a late fusion step is used to get a combined result set or the results of each modality are presented and kept separate [13, 14, 29]. Therefore, instead of fusing items into a final result set from which the top k are selected, the actual screen presenting the suggestions can be utilised to show the top $\frac{k}{M}$ results from each representation, where M is the number of modalities. Here **no fusion** (NF) takes place but rather the suggestions are divided into sections.An illustration of this method with 3 modalities can be seen in Figure 1b. If carefully implemented, this reduces computation as it avoids the aggregation step, and is more transparent with the user. This may also be beneficial in determining a dominant modality for a task.

A variation to this approach is a mix of rank aggregation and no fusion, where the final suggestion set consists of fusion slots (MM) and slots for each modality's top items (NF). We call this method **partial fusion** (PF). Figure 1c illustrates an example output of partial fusion with 3 modalities using 6 MM slots and 1 NF slot per modality. With a more diverse result set one might expect to get similar performance to rank aggregation in terms of computation, but potentially increase quality. Furthermore, this method may also be capable of determining a dominant modality for a task based on the selection of NF slots. With no fusion and partial fusion it is also important to consider the ordering of which items are presented to a user in a real use case, as that may influence the feedback they provide.

3 EXPERIMENTAL SETUP

To analyse the effects of late fusion with high-level features from one or more modalities from videos, we use three video datasets. In this section, we go over the user relevance feedback approach used in the experiments in 3.1, followed by a description of the evaluation protocols and their metrics in 3.2, and the details of the three datasets with regards to features, pre-processing, and tasks in 3.3-3.5.

3.1 Relevance Feedback System

We conduct the experiments using Exquisitor, which is an interactive learning system with user relevance feedback at its core. Exquisitor has shown state-of-the-art performance in large-scale interactive multimodal learning [18]. The approach compresses the multimedia representations of an item and stores them in approximate high-dimensional indexes. It consists of two phases; an offline phase and an online phase.

In the offline phase, high-level semantic features are extracted from the multimedia items. As these features are sparse, the top n features are selected and stored into a compact representation of 64-bit integers using the Ratio-64 compression from [32]. After getting the compressed representation a high-dimensional index is built for each modality/representation. Exquisitor uses a modified version of the extended Cluster Pruning (eCP) index. This cluster-based index is built by arbitrarily selecting a number of cluster representatives, like the first step in k-means, and then building a hierarchy from the representatives. The main point of the clusters is to be balanced to achieve response time guarantees. Once the index is built, Exquisitor is ready to proceed to the online phase.

The online phase of Exquisitor matches the general user relevance feedback approach. It loads the index and presents the user with an arbitrary set of items. After the user provides feedback on these items, the underlying classifier is trained. The classifier used by Exquisitor is linear SVM, which is well known for being efficient at training with few examples. Once trained, the resulting hyperplane is used to select the most relevant clusters (farthest from the hyperplane), followed by processing their items and presenting the top k. If multiple representations are involved, Exquisitor uses rank aggregation to perform late fusion. Due to the compression and high-dimensional indexing, Exquisitor is highly efficient with minimal resources, which allows it to allocate more resources when multiple representations are involved than other interactive learning approaches.

3.2 Evaluation Protocols and Metrics

To define automated evaluation for interactive learning we adhere to the principles of Analytic Quality (AQ) [33]. AQ is a means to define automated evaluation protocols for interactive learning approaches. With AQ, a set of artificial users label items from the provided suggestions. These users are referred to as actors. Each actor has one task, such as finding as many items fitting a description as possible, using the ground truth to label positive examples and treating everything else as negative. A total of s interactive sessions are performed with each actor consisting of r total rounds, where each session starts with a different starting point. This starting point is determined by supplying an initial p positive examples and n negative examples. There is also an option for supplying additional true negative examples along with the labeled examples each round.

To obtain results that confidently reflects the performance of a late fusion method, the number of interactive sessions and rounds for all protocols in this work is set to $s = 50$ and $r = 10$. The number of suggestions returned each interaction round is set to $k = 24$. While 10 rounds may seem as little, it is actually quite time consuming in the case of a real application. Assume the user thoroughly inspects each item, which takes 3-5 seconds. With 24 suggestions each round will take between 72-120 seconds, meaning 10 rounds takes between 12-20 minutes. It may be even more if the user not only views the keyframe of a shot but also plays the shot. For each interactive session, an actor starts with 3 positive examples from the ground truth and 5 negative examples (not in the ground truth) that are arbitrarily chosen. No additional true negatives are supplied during a session.

The metrics from these protocols are average precision per round, average recall after r rounds, and average latency per round. Since we are primarily focusing on the quality of late fusion methods we are interested in the precision and recall, and not response time, unless there is a noticeable impact. Note with 24 suggestions per round, a total of 240 items will be considered, which means large ground truth sets may have low recall numbers.

Regarding the weighted rank aggregation late fusion method, when a modality is preferred (W_{Mod}) it will have the weight set to 2.0 while any other modality involved will have a setting of 1.0. For the partial fusion method we perform the experiments with two settings for the number of non-fused slots per modality which are 2 (PF2) and 3 (PF3). For instance with 24 suggestions and 3 modalities, PF2 will have 18 MM slots and 2 NF slots per modality, and PF3 will have 15 MM slots with 3 NF slots per modality.

3.3 V3C1

V3C1 is a dataset consisting of 7,475 videos constituting 1,000 hours of footage. The videos are from the video sharing site Vimeo. The videos have textual metadata such as categories, tags, and descriptions. The individual videos range from 18 seconds to 1 hour, with an average length of 8 minutes. To be able to find shots from the videos that might be of interest for some task, we need to be able to extract features from smaller segments of the videos. Therefore, shot boundary detection is performed on the videos. As the shots may be of arbitrary lengths, we conform them to be within 1-10 seconds, leading to 1,007,360 total shots. From these shots we extract semantic concepts using a deep neural network that extracts 12,988 ImageNet concepts [21]. As multiple concepts can be present over the length of a shot, the representative feature vector chooses the highest scoring concepts across 1-5 frames depending on the segments length. We extract action related features from Kinetics-700 [5] using a 3D-ResNet model [12]. The actions are taken from the middle of the shot. Furthermore, we also extract features relating to the scene using a model for Places365 [35]. These are extracted from the middle frame of the shot. For all feature representations used in the experiments, the top 5 features are extracted and stored in the compressed representation.

V3C1 has been used in research as part of interactive retrieval challenges, namely TRECVID and Video Browser Showdown. These challenges focus on Known Item Search (KIS) tasks with regards to finding a specific shot within a video, correlating to either a textual description or a visual presentation of the shot to locate. Additionally, Ad-hoc Video Search (AVS) is another form of search task, where the intention is to find as many items matching a textual description. The tasks that we use in our experiments for the V3C1 dataset are from the 2021 TRECVID challenge on Ad-hoc Video Search. In total 20 queries are available with ground truth. The ground truth is built from real human assessors determining the relevance of submissions from participants. This implies not all items matching the query descriptions are in the ground truth. The number of relevant items for the actors range from 194 to 2,637.

3.4 Charades

Charades is a dataset made for research related to activity recognition of daily human activities. It consists of 9,848 videos of arbitrary length, albeit smaller than the previous dataset. Shot boundary detection has also been performed on this dataset, leading to 58,066 shots. Similar to the V3C1 dataset, we extract ImageNet concepts, Kinetics-700 actions, and scenes from Places365. The Charades dataset consists of a training and testing set. As we do not use the training set to train our model we include all items in the dataset.

The type of tasks related to this dataset are similar to AVS tasks where videos containing an activity need to be found. The Charades dataset has a total of 157 activity descriptions. Inspecting these descriptions shows that there are several small groups that are highly related in terms of concept and activity. We therefore randomly select 1 description from every 5, to get a varied selection of distinct tasks. Activities with fewer than 1,000 items are removed, leading to a total of 23 tasks for this dataset.

3.5 VGG-Sound

The last dataset we use is VGG-Sound, which is a large scale audio-visual dataset. The videos conveniently have a maximum length of 10 seconds. For this dataset we do not need to detect shots and can use the videos as is. Similarly to the previous two datasets the same features are extracted. Additionally we extract audio based concepts using an audio classification model [7].

We focus only on the testing set of this dataset, as the audio classification model has been trained with the training set. Out of the 15,446 test videos, only 9,996 videos have been obtained. The reason for missing videos is due to (i) the videos no longer being available, (ii) the videos having been made private, or (iii) API version issues.[1] The final dataset of VGG-Sound comes with 309

[1] The full list of videos that we have used along with the tasks of the different datasets can be found here: https://github.com/Ok2610/URF-VidMod.git

annotated classes divided amongst 9 categories. Instead of focusing on every class we select 3 from each category arbitrarily as the tasks for the evaluation. This leads to 27 varied tasks with relevant items ranging from 13 to 43.

4 RESULTS

In this section we present the results of the experiments conducted on each dataset using the different late fusion methods. The modality representations, as previously stated, are semantic concepts derived from ImageNet (Img), actions from Kinetics-700 (Act), scenes from Places365 (Scn), and audio from VGG-Sound (Aud).

4.1 TRECVID2021

Table 1 shows the results for the individual modalities, ImageNet, Actions, and Scenes, on the V3C1 dataset. The first column indicates average precision per round and the second column average recall after 10 rounds. Out of the three modalities, ImageNet performs far better than the others, in both precision and recall. With 24 items being displayed per round it indicates on average 3 of the items are from the ground truth. Note that the recall is low as the number of items covered over 10 rounds is not close to the number of total items of the ground truth. Actions is the second strongest, but its precision and recall are less than half of ImageNet. Scenes performs the worst out of the three. This is no real surprise as majority of the tasks from this challenge focus on events where concepts are more prevalent. While some tasks do refer to an action occurring in the video segment, these also highlight concepts which the ImageNet modality can use. As for Scenes it is evident that the tasks do not highlight the surroundings, with 4 tasks not finding any relevant items and other tasks not finding relevant items over multiple rounds.

Table 1: Results of individual modalities for TRECVID 2021.

Modality	Precision	Recall
Img	**0.131**	**0.057**
Act	0.056	0.023
Scn	0.037	0.018

Table 2: Average Precision per round (TRECVID2021)

	Rank	W_{Img}	W_{Act}	W_{Scn}	NF	PF(2)	PF(3)
Img+Act	0.144	**0.158**	0.120	-	0.094	0.146	0.144
Img+Scn	0.115	**0.130**	-	0.092	0.104	0.117	0.116
Act+Scn	**0.078**	-	**0.078**	0.074	0.068	0.075	0.075
All	0.137	**0.163**	0.125	0.112	0.137	0.141	0.140

Table 3: Average Recall after 10 rounds (TRECVID2021)

	Rank	W_{Img}	W_{Act}	W_{Scn}	NF	PF(2)	PF(3)
Img+Act	0.065	**0.071**	0.055	-	0.041	0.066	0.065
Img+Scn	0.049	**0.055**	-	0.041	0.045	0.051	0.050
Act+Scn	0.034	-	**0.035**	0.032	0.032	0.034	0.034
All	0.060	**0.071**	0.055	0.048	0.060	0.063	0.062

Table 2 shows the average precision of each modality combination with respect to the various methods for fusion. The first column indicates the precision of late fusion by rank aggregation. The following three columns are of weighted rank aggregation, when preferring ImageNet (W_{Img}), Actions (W_{Act}), and Scenes (W_{Scn}), respectively. The remaining columns show the precision for no fusion (NF), and partial fusion with 2 (PF2) and 3 (PF3) non-fused slots per modality. The highest average precision achieved is 0.163 by W_{Img} using all modalities. The lowest average precision of 0.068 comes from Act+Scn when used with NF. The reason for the low precision in general is due to the ground truth not covering the entire collection, as mentioned in Section 3.3. Looking into the modality combinations for the baseline rank aggregation, we observe Img+Act has a better precision than just using the ImageNet concepts. Img+Scn on the other hand has a lower precision, although still being over 3 times better than Scenes alone. Act+Scn has a higher precision than both Actions and Scenes individually. If all modalities are used with rank aggregation, the precision is higher than just ImageNet, but is lower than Img+Act. With regards to weighted rank aggregation, when it prefers the strongest modality, ImageNet, it achieves the highest precision of all approaches. Preferring Actions and Scenes is worse in all cases where ImageNet is present, but between W_{Act} and W_{Scn}, preferring Actions is better. If no fusion is performed and the result set is divided amongst the modalities, the precision for two modality combinations is lower than the baseline, but equal to it when all three modalities are used. As for the partial fusion with two or three non-fused slots, the precision is better than the baseline, with PF2 being better than PF3.

Table 3 shows the average recall of the modality combinations and fusion methods, where the columns are the same as Table 2. The results mimic the precision observations, with W_{Img} being best with a recall of 0.071 with all modalities involved and when excluding Scenes (Img+Act). The lowest recall of 0.032 is from no fusion with Act+Scn. While all fusion methods with Act+Scn achieve a higher average recall than using them individually. Comparing the recall of the lone ImageNet run with the combinations including ImageNet, higher recall is achieved with Img+Act and all modalities. Img+Scn performs worse, highlighting Scenes as a bad modality.

The results show that on average the best fusion method is weighted rank aggregation preferring ImageNet. In terms of modalities, the best average results are achieved when all modalities are involved. However, when considering individual tasks, only 7 out of the 20 tasks achieve higher precision and recall when compared to the results of Img+Act and Img+Scn for W_{Img}. Img+Act manages to have the most task improvements with 9 out of 20 tasks, while Img+Scn has 3 out of 20. The 7 tasks that score the highest for all modalities combined do insinuate elements for all three modalities. Take for instance the tasks “person wearing an apron indoors” (task 8) or “two boxers in a ring” (task 17). Here the concepts of “person”, “apron”, “boxer”, and “ring”, are useful for ImageNet. For Scenes and Actions one can infer a kitchen setting where a person is cooking for the former task, and an indoor boxing stadium/gym where people are boxing for the latter. For the tasks favoring Img+Act, the background scenery is not important, such as “person painting on a canvas” (task 10) or “man pointing his finger” (task 12). These tasks can take place anywhere, such as painting at a studio, outdoors, or in their home. For tasks favoring Img+Scn it is not necessarily that an action can not be associated, but it may be difficult to obtain. For instance, in the task “hang glider floating in the sky on a sunny day” (task 1) the concepts of “hang glider”, “sunny”, and “sky” are useful for ImageNet and Scenes, but the closest detectable action for this will be “paragliding”, which will introduce shots of parachutes rather than hang gliders.

Overall from the results we see when a bad modality is involved, it takes away from the stronger modalities. As expected, applying weights on the stronger modality, improves the results. No fusion performs worse, while partial fusion is the second best, which indicates that showing non-fused examples can be good and may be beneficial towards dynamic weight adjustments.

Table 4: Results of individual modalities for Charades

Modality	Precision	Recall
Img	**0.095**	**0.007**
Act	0.089	0.006
Scn	0.081	0.005

Table 5: Average Precision per round (Charades)

	Rank	W_{Img}	W_{Act}	W_{Scn}	NF	PF(2)	PF(3)
Img+Act	0.121	0.123	0.111	-	0.100	**0.124**	**0.124**
Img+Scn	0.099	0.105	-	0.095	0.097	**0.106**	**0.106**
Act+Scn	0.108	-	0.109	0.107	0.103	**0.111**	**0.111**
All	**0.131**	0.130	0.125	0.119	0.126	0.128	0.127

Table 6: Average Recall after 10 rounds (Charades)

	Rank	W_{Img}	W_{Act}	W_{Scn}	NF	PF(2)	PF(3)
Img+Act	0.009	0.009	0.008	-	0.007	**0.010**	**0.010**
Img+Scn	0.007	0.007	-	0.007	0.007	**0.009**	**0.009**
Act+Scn	0.008	-	0.008	0.007	0.007	**0.009**	**0.009**
All	0.009	0.009	0.009	0.009	0.009	**0.010**	**0.010**

4.2 Charades

Table 4 shows the average precision and recall of the individual modalities on the Charades dataset. Similar to the TRECVID evaluation, ImageNet is the strongest modality, but the difference towards the other modalities is less. Again, the recall is extremely low due to the size of the ground truth set. The videos in this dataset focus more on actions from humans which does show some tasks being better with the Action modality. Scenes is capable of finding relevant items in all tasks, albeit being worse at it than the others.

Table 5 shows the average precision for the different modality combinations and fusion methods. For this dataset the highest precision is 0.131 and is achieved by rank aggregation with all three modalities. The lowest precision is 0.095 by W_{Scn}. In the case of the baseline rank aggregation we see an increase in precision across all combinations. One noticeable result is that Scenes interferes less with Actions than ImageNet, as seen by the precision of Img+Act and Act+Scn. The highest precision is achieved by using all three modalities. Adding weights on ImageNet further improves the precision when two modalities are involved, however, when all three are used it is slightly lower than the baseline. Weights on Actions improve the precision only when used with Scenes, while weights on Scenes is always lower than the baseline. No fusion shows improvement over the individual modalities, but is not better than the baseline fusion. The partial fusion approach achieves the best precision for two modality combinations.

Table 6 shows the average recall for the tasks in Charades. The highest recall of 0.010 is observed from the partial fusion methods with ImageNet and Actions and all three modality representations. The lowest recall is 0.007 from no fusion with ImageNet and Scene. The recall is far lower for Charades as the ground truth is significantly larger, and again there is the potential of false negatives. With the results being similar to the precision, aside from the highest precision which is from rank aggregation with all modalities, it is safe to proclaim that overall partial fusion is better for this dataset, with 2 non-fusion slots being better than 3 again, though the difference is minuscule.

Surprisingly, for this dataset, setting weights on the strongest modality does not achieve the best results. It does improve over the baseline rank aggregation, but fails to beat partial fusion. So having a user set weights is still preferable over the baseline, but may not be a necessity to achieve better results. With regards to the performance of the modality combinations for PF2, the same phenomenon is observed as TRECVID results. Some tasks perform better with all modalities while others are better when one modality is excluded. Of the 23 tasks, 9 perform best with all modalities, 10 favor Img+Act, and 4 are better with Img+Scn. The tasks where all modalities work well have descriptions where the multiple modalities can benefit each other, such as “throwing something on the floor” (task 18), where the “floor” leads to indoor Scenes, “throwing” fits Actions, and “something” allows any object based concept from ImageNet to be involved. For Img+Act the tasks do not care for the scenery of the relevant shots, such as “holding a phone/camera” (task 2) and “watching/reading/looking at a book” (task 5). For these tasks the Scenes modality can steer the multimodal results in unrelated directions, while also taking additional non-fused slots due to the partial fusion method. The 4 tasks favoring Img+Scn are fairly open in terms of scenery, but the concepts involved make it easy to infer specific environments. For instance the task “putting something on a shelf somewhere”, while open-ended, with the concept of “shelf” indoor Scenes of office or home environments can easily be applied.

4.3 VGG-Sound

Recall that the tasks of VGG-Sound focus on events occurring in videos based on audible cues. Here we perform experiments with Audio features as well. Majority of the tasks focus on sounds coming from objects or from an entity performing an action, with few relating to sounds from the scenery. As such, the results including Scenes show similar traits to the TRECVID results: as the lone modality it performs poorly, and interferes with other modalities to make results worse. Hence, Scenes are not discussed further here.

Table 7 shows the average precision and recall for the individual modalities. As the ground truth sets for tasks in this dataset are much smaller in comparison to the other datasets, the recall is much higher. For this dataset the Audio dominates over ImageNet and Actions. ImageNet and Actions are close in performance, with Actions being slightly better.

Table 8 shows the average precision for the modality combinations and the different fusion methods. The highest precision is 0.074 and is achieved when all modalities are involved using weighted rank aggregation on the clearly dominant representation for Audio. The lowest precision of 0.036 is from Img+Act used with no fusion. The baseline rank aggregation improves the performance of ImageNet and Actions over their individual run. For all combinations with Audio, the other modalities act more as an interference, leading to worse performance than with just Audio. When weights are applied on each modality, it is clear that when put on a clear strong modality the results improve overall. Weights on the other modalities perform worse than the baseline results. With no fusion we observe that it improves over the baseline for Action and Audio, and manages to have better precision than the individual Audio run with all three modalities involved. This shows that while the other modalities may not be as strong as Audio, they are still good enough to provide relevant items. Looking at the two partial fusion methods results they manage to get the highest precision when only ImageNet and Actions are used. Img+Aud and Act+Aud manage to get better or similar precision as the individual Audio run.

Table 9 shows the average recall for the fusion methods and modality combinations. We observe a similar pattern as the precision results. W_{Aud} achieves the best recall where Audio is involved, highest being 0.557, and the best recall for Img+Act is achieved by partial fusion with 2 non-fused slots. While the precision for no fusion with all modalities is higher than the solo Audio precision, the recall is less. Similarly while the partial fusion methods achieve the same precision as the individual Audio run, the recall is better with all combinations. With a clear strong modality the 3 non-fusion slot per modality does perform better than 2 slots in some cases.

Inspecting the performance of tasks for the W_{Aud} modality combinations, we again observe that with all modalities a higher average precision and recall is achieved, but for a number of the tasks excluding a modality is better. Specifically, only 8 of the 27 tasks achieve their best performance with all modalities present, 13 tasks are better with Img+Aud, while the remaining 6 tasks are better with Act+Aud. Tasks such as “lawn mowing” (task 2), “basketball bounce” (task 4), and “canoe, kayak rowing” (task 14) work well with all modalities as they have a fair representation for certain elements of the descriptions. For Img+Aud, the task do not pertain any major action, but rather a concept/object which makes a sound, such as “train horning” (task 1) and “chinchilla barking” (task 6). For the tasks that favor Act+Aud, the concepts involved may not always fit the description, such as “playing squash” (task 3), where a concept such as “tennis” may lead to wrong results, while the action of “playing tennis” along with Audio for “playing squash” will better help each other.

Table 7: Results of individual modalities for VGG-Sound

Modality	Precision	Recall
Img	0.028	0.195
Act	0.029	0.207
Aud	**0.069**	**0.521**

Table 8: Average Precision per round (VGGSound)

	Rank	W_{Img}	W_{Act}	W_{Aud}	NF	PF(2)	PF(3)
Img+Act	0.040	0.040	0.039	-	0.036	**0.041**	**0.041**
Img+Aud	0.068	0.059	-	**0.073**	0.065	0.070	0.069
Act+Aud	0.062	-	0.052	**0.071**	0.067	0.070	0.070
All	0.066	0.063	0.056	**0.074**	0.072	0.070	0.070

Table 9: Average Recall after 10 rounds (VGGSound)

	Rank	W_{Img}	W_{Act}	W_{Aud}	NF	PF(2)	PF(3)
Img+Act	0.286	0.283	0.279	-	0.256	**0.291**	0.290
Img+Aud	0.514	0.443	-	**0.554**	0.486	0.530	0.522
Act+Aud	0.464	-	0.384	**0.540**	0.504	0.527	0.530
All	0.492	0.466	0.410	**0.557**	0.506	0.527	0.533

4.4 Discussion

From the experiments conducted on the different datasets, it is apparent that when more modalities are present fusion is important. This is an observation from all the datasets, where the no fusion results fare worse than the other methods. It is also evident that fusion with all modalities leads to higher precision and recall on average, however, this does not mean it is better for all tasks. This is highlighted in every dataset, where different combinations of modalities lead to different tasks performing better. Hence, the choice of modalities for a task matters. Furthermore, if it is clear which modality a task favors, a preference can be set for that modality, which can be extremely beneficial, as shown with the weighted rank aggregation results for TRECVID and VGG-Sound, with increases in average precision of 24% and 7%. In case it is not clear which modality to set a preference towards, showing mixed suggestions of fused items and a number of items from each modality, is more appropriate than regular fusion, or arbitrarily choosing a preference. This is evident from the Charades results, where partial fusion achieves better results than weighted rank aggregation. Aside from the takeaways relating to fusion, it is also worth noting that a modality’s performance in one dataset does not translate over to another. We have shown that the ImageNet modality is strong for TRECVID and Charades, which contain a wide variety of shots. However, its strength diminishes when dealing with VGG-Sound, which is not necessarily due to its representation, but is more related to the tasks focusing on more than simply a concept’s presence. The Action modality does not perform well for TRECVID, as the dataset contains many shots where someone is not performing an action. In Charades and VGG-Sound an action taking place is more common, hence, the improved performance.

5 CONCLUSION

In this paper we highlight the impact of different modality representations in videos for user relevance feedback with late fusion. We specifically analyse late fusion using rank aggregation, weighted rank aggregation, split modality output (no fusion), and partial fusion. Through experiments conducted on three video datasets, V3C1, Charades, and VGG-Sound, we show that modality fusion is generally beneficial, although weaker modalities can negatively impact the quality of the results. If the user knows beforehand which modality is better suited for a task, prioritising the results from that modality during fusion is greatly beneficial, while prioritising a bad modality for a task can be detrimental. Furthermore, diversifying the results with items coming from fusion and from individual modalities, is better than regular rank aggregation and no fusion. Finally, the performance and relation between modalities can change from one dataset to another.

ACKNOWLEDGMENTS

This work was supported by a PhD grant from the IT University of Copenhagen and by the European Regional Development Fund (project Robotics for Industry 4.0, CZ.02.1.01/0.0/0.0/15 003/0000470).

REFERENCES

[1] Fuad M Alkoot and Josef Kittler. 1999. Experimental evaluation of expert fusion strategies. *Pattern recognition letters* 20, 11-13 (1999), 1361–1369.

[2] Mutasem K Alsmadi. 2020. Content-based image retrieval using color, shape and texture descriptors and features. *Arabian Journal for Science and Engineering* 45, 4 (2020), 3317–3330.

[3] Miguel Arevalillo-Herráez, Juan Domingo, and Francesc J Ferri. 2008. Combining similarity measures in content-based image retrieval. *Pattern Recognition Letters* 29, 16 (2008), 2174–2181.

[4] Fabian Berns, Luca Rossetto, Klaus Schoeffmann, Christian Beecks, and George Awad. 2019. V3C1 Dataset: An Evaluation of Content Characteristics *(ICMR '19)*. Association for Computing Machinery, New York, NY, USA, 334–338.

[5] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. 2019. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987* (2019).

[6] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. 2020. Vggsound: A Large-Scale Audio-Visual Dataset. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. IEEE, 721–725. https://doi.org/10.1109/ICASSP40776.2020.9053174

[7] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. 2020. VGGSound: A Large-scale Audio-Visual Dataset. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[8] Shizhe Chen and Qin Jin. 2016. Multi-Modal Conditional Attention Fusion for Dimensional Emotion Prediction. In *Proceedings of the 24th ACM International Conference on Multimedia* (Amsterdam, The Netherlands) *(MM '16)*. Association for Computing Machinery, New York, NY, USA, 571–575.

[9] Minh-Son Dao, Pham Quang Nhat Minh, Asem Kasem, and Mohamed Saleem Haja Nazmudeen. 2018. A Context-Aware Late-Fusion Approach for Disaster Image Retrieval from Social Media. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval* (Yokohama, Japan) *(ICMR '18)*. Association for Computing Machinery, New York, NY, USA, 266–273.

[10] Hugo Jair Escalante, Carlos A Hérnadez, Luis Enrique Sucar, and Manuel Montes. 2008. Late fusion of heterogeneous methods for multimedia image retrieval. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*. 172–179.

[11] Ronald Fagin, Ravi Kumar, and Dandapani Sivakumar. 2003. Efficient similarity search and classification via rank aggregation. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. 301–312.

[12] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. 2018. Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6546–6555.

[13] Silvan Heller, Mahnaz Amiri Parian, Ralph Gasser, Loris Sauter, and Heiko Schuldt. 2020. Interactive lifelog retrieval with vitrivr. In *Proceedings of the Third Annual Workshop on Lifelog Search Challenge*. 1–6.

[14] Silvan Heller, Rahel Arnold, Ralph Gasser, Viktor Gsteiger, Mahnaz Parian-Scherb, Luca Rossetto, Loris Sauter, Florian Spiess, and Heiko Schuldt. 2022. Multi-modal interactive video retrieval with temporal queries. In *International Conference on Multimedia Modeling*. Springer, 493–498.

[15] Mohamed Maher Ben Ismail. 2017. A survey on content-based image retrieval. *International Journal of Advanced Computer Science and Applications* 8, 5 (2017).

[16] Björn Þór Jónsson, Omar Shahbaz Khan, Dennis C Koelma, Stevan Rudinac, Marcel Worring, and Jan Zahálka. 2020. Exquisitor at the Video Browser Showdown 2020. In *Proc. MMM*. Springer, 796–802.

[17] Antonio Juárez-González, Manuel Montes-y Gómez, Luis Villaseñor-Pineda, David Pinto-Avendaño, and Manuel Pérez-Coutiño. 2010. Selecting the n-top retrieval result lists for an effective data fusion. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 580–589.

[18] Omar Shahbaz Khan, Björn Þór Jónsson, Stevan Rudinac, Jan Zahálka, Hanna Ragnarsdóttir, Þórhildur Þorleiksdóttir, Gylfi Þór Guðmundsson, Laurent Amsaleg, and Marcel Worring. 2020. Interactive Learning for Multimedia at Large. In *European Conference on Information Retrieval*. Springer.

[19] Kuan-Ting Lai, Dong Liu, Shih-Fu Chang, and Ming-Syan Chen. 2015. Learning Sample Specific Weights for Late Fusion. *IEEE Transactions on Image Processing* 24, 9 (2015), 2772–2783.

[20] František Mejzlík, Patrik Veselý, Miroslav Kratochvíl, Tomáš Souček, and Jakub Lokoč. 2020. SOMHunter for Lifelog Search. In *Proc. of the 3rd Annual Workshop on Lifelog Search Challenge*. ACM, 73–75.

[21] Pascal Mettes, Dennis C. Koelma, and Cees G.M. Snoek. 2016. The ImageNet Shuffle: Reorganized Pre-Training for Video Event Detection *(ICMR '16)*. Association for Computing Machinery, New York, NY, USA, 175–182.

[22] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. 2021. Attention bottlenecks for multimodal fusion. *Advances in Neural Information Processing Systems* 34 (2021), 14200–14213.

[23] Luca Piras and Giorgio Giacinto. 2017. Information fusion in content based image retrieval: A comprehensive overview. *Information Fusion* 37 (2017), 50–60.

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. *CoRR* abs/2103.00020 (2021). arXiv:2103.00020 https://arxiv.org/abs/2103.00020

[25] M Elena Renda and Umberto Straccia. 2003. Web metasearch: rank vs. score based rank aggregation methods. In *Proceedings of the 2003 ACM symposium on Applied computing*. 841–846.

[26] William Ribarsky and Brian Fisher. 2016. The Human-Computer System: Towards an Operational Model for Problem Solving. In *Proc. Hawaii Int. Conf. on Sys. Sciences*. 1446–1455.

[27] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ivan Laptev, Ali Farhadi, and Abhinav Gupta. 2016. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. *ArXiv e-prints* (2016). arXiv:1604.01753 http://arxiv.org/abs/1604.01753

[28] Cees GM Snoek, Marcel Worring, and Arnold WM Smeulders. 2005. Early versus late fusion in semantic video analysis. In *Proc. ACMMM*. 399–402.

[29] Xiao-Yong Wei, Yu-Gang Jiang, and Chong-Wah Ngo. 2011. Concept-driven multi-modality fusion for video search. *IEEE Transactions on Circuits and Systems for Video Technology* 21, 1 (2011), 62–73.

[30] Pengcheng Wu, Steven CH Hoi, Peilin Zhao, Chunyan Miao, and Zhi-Yong Liu. 2015. Online multi-modal distance metric learning with application to image retrieval. *ieee transactions on knowledge and data engineering* 28, 2 (2015), 454–467.

[31] Bo Yang, Tao Mei, Xian-Sheng Hua, Linjun Yang, Shi-Qiang Yang, and Mingjing Li. 2007. Online video recommendation based on multimodal fusion and relevance feedback. In *Proceedings of the 6th ACM international conference on Image and video retrieval*. 73–80.

[32] Jan Zahálka, Stevan Rudinac, Björn Þór Jónsson, Dennis C Koelma, and Marcel Worring. 2018. Blackthorn: Large-Scale Interactive Multimodal Learning. *IEEE TMM* 20, 3 (2018), 687–698.

[33] Jan Zahálka, Stevan Rudinac, and Marcel Worring. 2015. Analytic Quality: Evaluation of Performance and Insight in Multimedia Collection Analysis. In *Proc. ACMMM*. ACM, 231–240.

[34] Jan Zahálka and Marcel Worring. 2014. Towards interactive, intelligent, and integrated multimedia analytics. In *Visual Analytics Science and Technology (VAST)*. IEEE, 3–12.

[35] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Places: A 10 million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).

Chapter 4

Demonstrations and Interactive Search Challenges

When designing an interactive learning approach, it is important to realise that it consists of two parts, a foundation for the interactive learning process and a client application that the human user interacts with. While the previous chapter covered the foundation, this chapter focuses on Exquisitor as a client application.

The basic interface for an interactive learning approach typically consists of the suggestions displayed in a grid, with buttons for labeling an item positive or negative. This constitutes the principles of the approach, but it is worth considering whether such interface is sufficient for exploring and searching through large multimedia collections. Section 4.1 presents a demonstration paper entitled **Exquisitor: Breaking the Interaction Barrier for Exploration of 100 Million Images**, which has been published in the proceedings of 27th ACM Multimedia conference (ACMMM 2019) [82]. In this demonstration, the initial version of Exquisitor from Section 3.1 is used to interact with the YFCC100M collection on a laptop, using a web interface following the design described above. As the intention of this demonstration was on exploring the collection, additional functions were available in the client, which focused on getting items from a classifier with arbitrary examples, or an arbitrary set of items. From this demonstration, multiple observations were made. With regards to the presentation, simply asking people to explore a collection is not enough for all users, but rather giving them an initial objective of finding images containing a certain concept is better. The demonstration also confirmed that the research community has an interest in interactive learning to explore and search large collections, but responsive and accurate approaches such as Exquisitor are needed. As for the client application, it demonstrated that Exquisitor is able to find relevant items, and reliably work on a *single* laptop, where all computations are done, and achieves an average response time of 0.3 seconds. The only extension to the laptop was a 2 TB SSD to store the images of the YFCC100M collection. By showing that Exquisitor can run well on a laptop, has inspired research towards down scaling it to mobile phones [6]. In cases where users already had a goal in mind, starting from an arbitrary screen did lead to many rounds of labeling negatives or getting new screens with arbitrary items or concepts. From this observation, considerations towards adding filters, or a function to find initial positive examples arose.

To further understand the capabilities of Exquisitor with regards to search oriented tasks and tasks with shifting descriptions, it is important to introduce methods to reduce the search space in large collections. This has two effects, first it reduces computation and second it can help find relevant items faster. Section 4.2 and 4.3 presents two workshop papers for Exquisitor participating at the Lifelog Search Challenge (LSC) 2019 and 2020 at the International Conference of Multimedia Retrieval (ICMR) [52, 55]. Lifelog data consists of personal information relating to an individual, collected through smart gadgets. One of these gadgets is a miniature body-camera that takes images at set intervals, leading to a large image collections. At LSC, only subsets of the larger collections are used. The tasks presented at these challenges are textual known-item search tasks, where the goal is to find an item from the relevant item set. The task descriptions are presented live over a period of time, where additional details are provided or corrections are made, such as "Find an image of me in the office... It was a Tuesday... Actually it was a Thursday". The intention behind participating in these challenges is to analyse the properties of Exquisitor when solving very specific search-oriented tasks. The expectations are therefore not to get a high placement, but learning the benefits and shortcomings of interactive learning in a “real” setting.

During the first participation in this live interactive search challenge, Section 4.2, the Exquisitor client application consisted of the interface from the demonstration, along with filters for the images’ metadata information. Additionally, a function for getting a new suggestion set was added, which used the current classifier to get the next top ranked items. With the task descriptions focusing on metadata and visual aspects of the images, representations for both modalities were used. From the participation, it was evident that Exquisitor was capable of solving tasks that lean towards search, but improvements to the client application would be needed. One of these improvements was screen utilisation, as the column showing statistics of Exquisitor, takes space away from either enlarging images or presenting more images per round. Furthermore, not having a method to quickly find some positive examples, meant initial interaction rounds being spent on negative labeling. In the LSC 2020 participation, in Section 4.3, all of these improvements were made, by adding a keyword search for concepts to find initial positive examples and improving screen space. From these improvements, Exquisitor is capable of solving tasks in image collections. From the second participation, many task descriptions indicated the presence of exact text within images, which Exquisitor did not have any representations for, nor did it have filters for such, showing that different tasks may highlight different modalities. Even then, Exquisitor managed to solve some of the tasks, but it is important to consider additional filters and modalities for the interactive learning approach, to better support realistic tasks.

While participation in LSC shows that the Exquisitor client works for images, multimedia collections also consists of videos. To examine the application with regards to videos and tasks focusing on finding events occurring within videos, Exquisitor has participated in the Video Browser Showdown (VBS) 2020 and 2021. Section 4.4 and 4.5 present the workshop papers for VBS 2020 and 2021 which are workshops held at the Multimedia Modeling conference (MMM) [49, 51]. These challenges are similar to LSC, but revolve around a video collection and have been going for over a decade now. Similar to LSC, it has textual known item search (T-KIS) tasks, along with visual known item search (V-KIS) where the relevant video segment to find is played live, and ad-hoc video search (AVS) tasks which focus on finding as many segments as possible matching a description. VBS is done over two sessions, one involving the expert users and one involving the novice user. The novice user is a person selected from the audience at the workshop. It should also be noted that the VBS challenges allow up to two users, which for Exquisitor meant two users running Exquisitor in isolation on their own laptops.

The first participation showed that Exquisitor is capable of handling tasks for video collections. However, Exquisitor struggled with task descriptions that had a focus on temporal aspects of a segment, as there is no notion for this in the foundation or the client application. From the novice session, it became clear that a common user is still used to searching with keywords, as the search functionality intended for finding positive examples for the interactive learning process, was used more as a regular search tool. In the second participation, the temporal task descriptions were addressed by adding the ability to work with multiple models in the client and then use a merge function to find videos containing segments from both models. There were no novice session for VBS 2021, as the event had been made virtual due to COVID-19. Generally, observing all the users, expert or novice from VBS and LSC, highlighted how users with experience and expertise label, filter, and search with the client application, compared to novices. Additionally, as these workshops run over a long session, fatigue also plays a part, which can lead to users underperforming. These behavioral observations were not present in any of the automated evaluations, which is what motivated the study of their impact, leading to the article in Section 3.2. Furthermore, it became apparent that Exquisitor needed to have some form of incremental retrieval, as otherwise the scope of the index was set to process the entire collection. Fortunately, this caused no real performance issues due to the small collection sizes of LSC and VBS, but if it was a large-scale collection, the story would have been different, which motivated the research towards incremental retrieval and query optimisation policies presented in the journal article in Section 3.3. Furthermore, given that the task descriptions often involve colors of clothing, or actions performed by a person, it is worthwhile to add additional modality representations, and optimising the fusion based on the findings from the article in Section 3.4.

Through participating in the demonstration and workshops of LSC and VBS, the Exquisitor client has been shown to be well suited for performing exploration- and search-oriented tasks on multimedia collections. It is clear, however, that due to its interactive learning nature, retrieval systems that include multiple search functionalities will fare better at the challenges. However, due to the low resource utilisation of Exquisitor, additional search functionality is possible to introduce in the application, without hampering the interactive learning performance. Aside from the client application demonstrated at these venues, a mobile application following the principles of Exquisitor has also been developed [6, 90], as well as an alternative cross platform user interface [56]. Showcasing interactive learning on mobile devices is intriguing, as it is a new way of interacting with a collection which many are not accustomed to on smaller devices.

Exquisitor: Breaking the Interaction Barrier for Exploration of 100 Million Images

Hanna Ragnarsdóttir
Reykjavik University
Reykjavik, Iceland
hannar15@ru.is

Þórhildur Þorleiksdóttir
Reykjavik University
Reykjavik, Iceland
thorhildurt15@ru.is

Omar Shahbaz Khan
IT University of Copenhagen
Copenhagen, Denmark
omsh@itu.dk

Björn Þór Jónsson
IT University of Copenhagen
Copenhagen, Denmark
bjorn@itu.dk

Gylfi Þór Guðmundsson
Reykjavik University
Reykjavik, Iceland
gylfig@ru.is

Jan Zahálka
bohem.ai
Prague, Czech Republic
jan.zahalka@bohem.ai

Stevan Rudinac
University of Amsterdam
Amsterdam, Netherlands
s.rudinac@uva.nl

Laurent Amsaleg
CNRS–IRISA
Rennes, France
laurent.amsaleg@irisa.fr

Marcel Worring
University of Amsterdam
Amsterdam, Netherlands
m.worring@uva.nl

ABSTRACT

In this demonstration, we present Exquisitor, a media explorer capable of learning user preferences in real-time during interactions with the 99.2 million images of YFCC100M. Exquisitor owes its efficiency to innovations in data representation, compression, and indexing. Exquisitor can complete each interaction round, including learning preferences and presenting the most relevant results, in less than 30 ms using only a single CPU core and modest RAM. In short, Exquisitor can bring large-scale interactive learning to standard desktops and laptops, and even high-end mobile devices.

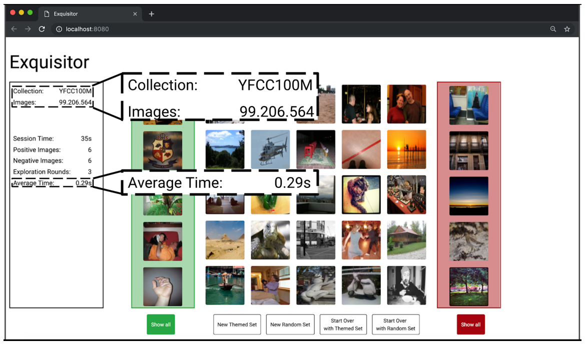
CCS CONCEPTS

• **Information systems** → **Multimedia and multimodal retrieval**; **Multimedia databases**;

KEYWORDS

Interactive multimodal learning; Scalability; 100 million images.

ACM Reference Format:
Hanna Ragnarsdóttir, Þórhildur Þorleiksdóttir, Omar Shahbaz Khan, Björn Þór Jónsson, Gylfi Þór Guðmundsson, Jan Zahálka, Stevan Rudinac, Laurent Amsaleg, and Marcel Worring. 2019. Exquisitor: Breaking the Interaction Barrier for Exploration of 100 Million Images. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3343031.3350580



Collection: YFCC100M
Images: 99.206.564

Session Time: 35s
Positive Images: 6
Negative Images: 6
Exploration Rounds: 3
Average Time: 0.29s

Figure 1: The Exquisitor demonstration interface. Exquisitor interactively learns new analytic categories over the full YFCC100M image collection, with average latency of less than 30 ms per interaction round, using hardware comparable with standard desktops and modern mobile devices.

1 INTRODUCTION

Multimedia collections have become a cornerstone data resource in a variety of scientific and industrial fields. One of the most difficult challenges for interactive exploration of such collections—not only for data scientists working with these collections, but also for the multimedia community as a whole—is their *scale*. How can we facilitate efficient access to multimedia collections comprising tens or hundreds of millions of images, let alone billions?

Interactive learning, which was embraced by the multimedia community in the early days of content-based image and video retrieval [4, 12], has recently experienced revival as an umbrella method capable of satisfying a variety of multimedia information needs, ranging from exploratory browsing to seeking a particular known item [17]. Through feedback from the user, interactive learning can adapt to the intent and knowledge of the user, and thus *collaborate* with the user towards, e.g., learning new or unknown analytic categories on the fly. Previous contributions, however, largely operated at a relatively small scale [1, 2, 8, 10, 13, 14].

MM '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10.
https://doi.org/10.1145/3343031.3350580

Consider, as an example, the YFCC100M collection [16], comprising 99.2M images and 0.8M videos. It has existed for some time now, but very few have a good idea about what its actual contents are. This is no surprise, as it is difficult to tackle this collection with existing techniques: the simple metadata-based filtering approach is impeded by the sparse and noisy nature of the metadata and, at this scale, similarity search is akin to shooting in the dark. Semantic concept detectors can be used to generate additional content-based metadata for both approaches, but that does not alleviate their problems. And thus, the YFCC100M collection remains a mystery.

We have recently developed Exquisitor, a highly scalable interactive multimodal learning approach [6]. A key feature that sets Exquisitor apart from related approaches is its scalability: Exquisitor can retrieve suggestions from 100 million images with sub-second latency, using extremely modest computing resources, thus breaking the interaction barrier for large-scale interactive learning. In this demonstration, we propose to allow ACM Multimedia attendees to interactively explore the YFCC100M collection with Exquisitor.

2 EXQUISITOR INTERFACE

The current Exquisitor user interface, shown in Figure 1, is browser-based and implemented using the React JavaScript library. It is a fairly traditional interactive learning interface, in that users are asked to label positive and negative examples, which are then used to learn their preferences and determine the new round of suggestions. Due to the extreme efficiency of the interactive learning process, however, there are some notable differences from traditional interactive learning interfaces:

- The learning process runs unobtrusively in the background, continuously providing new on-demand relevant examples as the user progresses with her exploration, instead of requiring explicit management.
- Individual images are replaced, rather than the entire screen, for a smooth transition from one interaction round to the other. Users are allowed to indicate that images are neither positive nor negative to get a new suggestion, and images that have been visible for some time, but not tagged as positive or negative, can also be replaced with new suggestions.
- Users can revisit positive and negative examples, removing or even reversing the feedback label, as their understanding of the collection contents and its relevance evolves.

Overall, the user interface is intended to provide a smooth learning experience. We have already used Exquisitor in the Lifelog Search Challenge (LSC) 2019 [7], and a detailed evaluation of the user experience is part of our future work.

3 THE LEARNING PROCESS

Exquisitor's back-end produces relevant results to show to the user in less than 30 ms per interaction round, including learning the user preference and scoring the collection, using one 2.4 GHz CPU core and less than 6 GB of RAM. The back-end system is composed of two web services, as shown in Figure 2. The ImageAPI service serves thumbnails for the YFCC100M image collection, as required by the user interface. The LearningAPI service wraps the underlying multimodal learning engine, decribed in [6]. In the remainder of this section, we briefly outline the multimodal learning process.

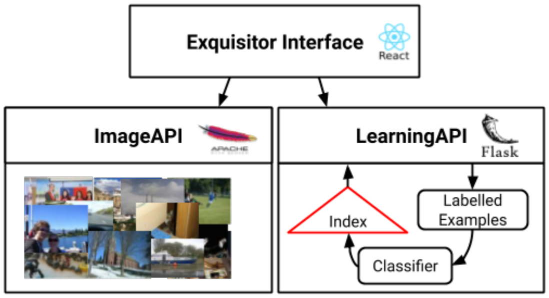


Figure 2: Exquisitor demonstration system overview.

To enable interactive multimodal learning, visual and text features were extracted from the images of the YFCC100M collection. For visual features, the 1000 ImageNet semantic concepts were extracted using a GoogLeNet architecture [15]. For the text modality, 100 LDA topics were extracted from the image title, tags and description using the gensim framework [11].

Uncompressed features for the 99.2M images require nearly 880GB of memory. Exquisitor uses the recently proposed Ratio-64 representation, which preserves only the top visual concepts and text topics for each image [18]. The compressed feature collections require less than 6GB of storage, thus fitting into the memory of a standard consumer PC, as well as some high-end mobile devices. This effective compression method has been shown to preserve the semantic descriptiveness of the visual and text features [18].

The interactive learning process is facilitated using a linear SVM model, proven to provide a good balance between efficiency and accuracy when classifying large datasets based on few training examples [5, 9]. Based on the relevance indication provided by the user, a classifier is trained separately for the text and visual modalities and the images furthest from the hyperplane are selected. The final list of results is created using rank aggregation.

The compressed feature data is indexed using a variant of the extended Cluster Pruning (eCP) high-dimensional indexing algorithm [3]. By directing Exquisitor's attention to the clusters with representatives most relevant to the learned linear SVM model, the work of scoring candidates is reduced by nearly two orders of magnitude, with an actual increase in quality [6]. The combination of all these state-of-the-art methods enables an interaction round of less than 30 ms on average using only limited computational resources: one 2.4 GHz CPU core and less than 6 GB of RAM.

4 DEMONSTRATION

The main emphasis of the demonstration will be to allow conference participants to explore the 99.2 million images of the YFCC100M collection using Exquisitor. The authors will also prepare some interesting exploration scenarios that highlight aspects of the YFCC100M collection. During the demonstration we hope to engage conference participants in a discussion that can inspire the multimedia community to work on scalable multimedia techniques and applications.

REFERENCES

[1] William H. Beluch, Tim Genewein, Andreas Nürnberger, and Jan M. Köhler. 2018. The Power of Ensembles for Active Learning in Image Classification. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Salt Lake City, UT, USA, 9368–9377.

[2] Kashyap Chitta, Jose M. Alvarez, and Adam Lesnikowski. 2018. Large-Scale Visual Active Learning with Deep Probabilistic Ensembles. arXiv:1811.03575. (2018), 10 pages.

[3] Gylfi Þór Guðmundsson, Björn Þór Jónsson, and Laurent Amsaleg. 2010. A Large-scale Performance Study of Cluster-based High-dimensional Indexing. In *Proc. International Workshop on Very-large-scale Multimedia Corpus, Mining and Retrieval (VLS-MCMR)*. ACM, Firenze, Italy, 31–36.

[4] Thomas S. Huang, Charlie K. Dagli, Shyamsundar Rajaram, Edward Y. Chang, Michael I. Mandel, Graham E. Poliner, and Daniel P. W. Ellis. 2008. Active Learning for Interactive Multimedia Retrieval. *Proc. IEEE* 96, 4 (2008), 648–667.

[5] Prateek Jain, Sudheendra Vijayanarasimhan, and Kristen Grauman. 2010. Hashing Hyperplane Queries to Near Points with Applications to Large-Scale Active Learning. In *Proc. Conference on Neural Information Processing Systems (NIPS)*. Curran Associates, Inc., Vancouver, BC, Canada, 928–936.

[6] Björn Þór Jónsson, Omar Shahbaz Khan, Hanna Ragnarsdóttir, Þórhildur Þorleiksdóttir, Jan Zahálka, Stevan Rudinac, Gylfi Þór Guðmundsson, Laurent Amsaleg, and Marcel Worring. 2019. Exquisitor: Interactive Learning at Large. arXiv:1904.08689. (2019), 10 pages.

[7] Omar Shahbaz Khan, Björn Þór Jónsson, Jan Zahálka, Stevan Rudinac, and Marcel Worring. 2019. Exquisitor at the Lifelog Search Challenge 2019. In *Proceedings of the ACM Workshop on Lifelog Search Challenge, LSC@ICMR 2019*. ACM, Ottawa, ON, Canada, 7–11.

[8] Akshay Mehra, Jihun Hamm, and Mikhail Belkin. 2018. Fast Interactive Image Retrieval using Large-Scale Unlabeled Data. arXiv:1802.04204. (2018), 15 pages.

[9] Ionuţ Mironică, Bogdan Ionescu, Jasper Uijlings, and Nicu Sebe. 2016. Fisher Kernel Temporal Variation-based Relevance Feedback for Video Retrieval. *Computer Vision and Image Understanding* 143 (2016), 38 – 51.

[10] Karl Ni, Roger A. Pearce, Kofi Boakye, Brian Van Essen, Damian Borth, Barry Chen, and Eric X. Wang. 2015. Large-Scale Deep Learning on the YFCC100M Dataset. arXiv:1502.03409. (2015), 5 pages.

[11] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, Valletta, Malta, 45–50.

[12] Yong Rui, Thomas S. Huang, and Sharad Mehrotra. 1997. Content-Based Image Retrieval with Relevance Feedback in MARS. In *Proc. International Conference on Image Processing (ICIP)*. IEEE Computer Society, Santa Barbara, CA, USA, 815–818.

[13] Nicolae Suditu and François Fleuret. 2012. Iterative relevance feedback with adaptive exploration/exploitation trade-off. In *Proc. ACM International Conference on Information and Knowledge Management (CIKM)*. ACM, Maui, HI, USA, 1323–1331.

[14] Nicolae Suditu and François Fleuret. 2016. Adaptive Relevance Feedback for Large-Scale Image Retrieval. *Multimedia Tools Appl.* 75, 12 (2016), 6777–6807.

[15] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. 2015. Going Deeper with Convolutions. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Boston, MA, USA, 1–9.

[16] Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. 2016. YFCC100M: The New Data in Multimedia Research. *Commun. ACM* 59, 2 (2016), 64–73.

[17] Jan Zahálka and Marcel Worring. 2014. Towards Interactive, Intelligent, and Integrated Multimedia Analytics. In *Proc. IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE Computer Society, Paris, France, 3–12.

[18] Jan Zahálka, Stevan Rudinac, Björn Þór Jónsson, Dennis C. Koelma, and Marcel Worring. 2018. Blackthorn: Large-Scale Interactive Multimodal Learning. *IEEE Transactions on Multimedia* 20, 3 (2018), 687–698.

Exquisitor at the Lifelog Search Challenge 2019

Omar Shahbaz Khan
IT University of Copenhagen
Copenhagen, Denmark
omsh@itu.dk

Björn Þór Jónsson
IT University of Copenhagen
Copenhagen, Denmark
bjorn@itu.dk

Jan Zahálka
bohem.ai
Prague, Czech Republic
jan.zahalka@bohem.ai

Stevan Rudinac
University of Amsterdam
Amsterdam, Netherlands
s.rudinac@uva.nl

Marcel Worring
University of Amsterdam
Amsterdam, Netherlands
m.worring@uva.nl

ABSTRACT

Interactive learning is an umbrella term for methods that attempt to understand the information need of the user and formulate queries that satisfy that information need. We propose to apply the state of the art in interactive multimodal learning to visual lifelog exploration and search, using the Exquisitor system. Exquisitor is a highly scalable interactive learning system, which uses semantic features extracted from visual content and text to suggest relevant media items to the user, based on user relevance feedback on previously suggested items. Findings from our initial experiments indicate that interactive multimodal learning will likely work well for some LSC tasks, but also suggest some potential enhancements.

CCS CONCEPTS

• **Information systems → Multimedia and multimodal retrieval**; **Multimedia databases**.

KEYWORDS

Lifelogging; Interactive learning; Exquisitor.

ACM Reference Format:
Omar Shahbaz Khan, Björn Þór Jónsson, Jan Zahálka, Stevan Rudinac, and Marcel Worring. 2019. Exquisitor at the Lifelog Search Challenge 2019. In *Lifelog Search Challenge'19 (LSC'19), June 10–13, 2019, Ottawa, ON, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3326460.3329156

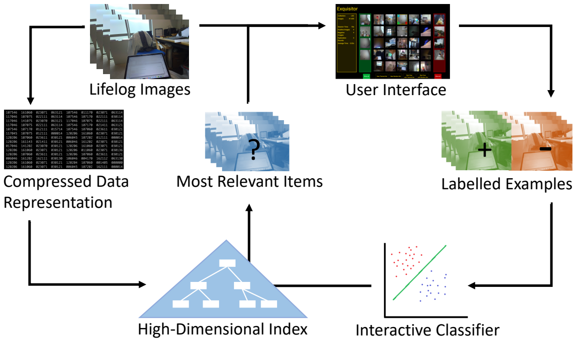


Figure 1: Exquisitor's interactive learning pipeline. Initially, the lifelog's image collection is processed to produce a compressed semantic representation, that is stored in a scalable high-dimensional index. In each round of the interactive learning process, the user is shown a set of potentially relevant images. The user's judgments are then used to train a classifier, which in turn is used to retrieve a new set of images to show to the user. With the LSC collection, producing new suggestions takes about 30ms on a laptop computer.

1 INTRODUCTION

Today's plethora of small devices allows capturing a tremendous amount of personal information. The people who make use of these devices to the fullest extent, gathering a variety of information about their daily lives, are termed lifeloggers. The most important feature of a typical *lifelog* is the image collection generated by a camera attached to the individual lifelogger taking pictures at regular intervals. The lifelog can also contain other sensor data, such as temperature, location, heart rate, and audio, depending on which devices the individual uses. Furthermore, this data can be processed with state-of-the-art computer vision and learning algorithms to produce semantic annotations. Applications of such personal lifelog data include self-monitoring and assisted memory [10].

The Lifelog Search Challenge (LSC) is a competition where researchers are asked to study and develop methods to solve search-related tasks for a multimodal lifelog dataset. Each task in LSC is an independent query, to be solved in a few minutes, where a correct result is a single image returned from a set of relevant images. The query description is given gradually, as might be typical when a lifelog is used to find information and the user slowly remembers more details about the situation. The first edition of LSC, held in 2018, showcased a variety of multimedia browsers aiming to search the lifelog with different approaches, ranging from traditional keyword search to novel virtual reality-based approaches [8].

Working with a lifelog should be a highly interactive process, where the lifelog user is collaborating with the lifelog system on a variety of tasks, ranging from pure exploration of the lifelog collection to focused search tasks to retrieve images relating to

LSC'19, June 10–13, 2019, Ottawa, ON, Canada

ACM ISBN 978-1-4503-6781-3/19/06…$15.00
https://doi.org/10.1145/3326460.3329156

particular memories. Multimedia analytics has been proposed as a research area aimed exactly at solving such diverse interactive information needs [23]. In multimedia analytics, an analytical session is composed of multiple different sub-tasks, ranging from browsing to seeking a particular known item, thus forming an exploration-search axis. Furthermore, *interactive multimodal learning* was proposed as an umbrella task capable of satisfying all the tasks on the exploration-search axis [23]. It is therefore of significant interest to apply interactive multimodal learning to LSC.

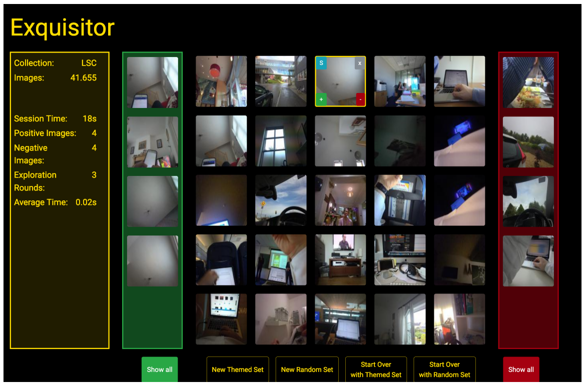
We have recently developed Exquisitor, a highly scalable interactive multimodal learning approach [13]. Figure 1 illustrates the iterative feedback process employed by Exquisitor as employed with lifelog data. When a lifelog user has an information need, she is initially presented with a set of randomly selected images from the lifelog and asked to give feedback on (some of) the items. The feedback is used to build (and subsequently update) a classification model, which in turn is used to provide new suggestions; this iterative process continues as long as the user deems necessary. A key feature that sets Exquisitor apart from other interactive learning approaches is its scalability: Exquisitor can retrieve suggestions from the YFCC100M collection with sub-second latency, using computing resources that are comparable to today's high-end mobile device. In this paper, we propose to use Exquisitor to solve the tasks of the Lifelog Search Challenge.

The remainder of the paper is organized as follows. In Section 2, we briefly give background for interactive learning and LSC. Section 3 then outlines the Exquisitor approach and its exploration interface. In Section 4, we look at the dataset provided by LSC and describe the processing required to use it with the Exquisitor approach. In Section 5, we briefly report on initial experiments with interactive retrieval tasks, before concluding the paper in Section 6.

2 BACKGROUND

Interactive learning comes in two basic forms, *active learning* and *user relevance feedback* [11]. In active learning, the goal is to create the best possible classifier, so the contribution of the user is typically to annotate samples close to the decision boundary between classes [2, 12]. User relevance feedback algorithms, in contrast, focus on giving users insight into the multimedia collections [17]. As a result, relevance feedback systems typically present as suggestions to the user the items for which the classification model is the most confident [19]. While this latter strategy may require more interactions to achieve the same final quality of the classification model, users may achieve their desired knowledge earlier [23].

Originally proposed in the 90s, early user relevance feedback systems for content-based image and video retrieval commonly relied on visual features that lack meaningful representation, such as colour, texture, shape and edge histograms [19], as well as indexing techniques that are inefficient in high-dimensional spaces, such as R-trees and kd-trees [4]. While relatively little work has been done on user relevance feedback in the last decade, recent advances in both high-dimensional indexing and data representation, along with calls for action from the multimedia community [21, 23], have motivated us to re-visit user relevance feedback with the Exquisitor approach [13].



Collection: LSC
Images: 41.655

Session Time: 18s
Positive Images: 4
Negative Images: 4
Exploration Rounds: 3
Average Time: 0.02s

Figure 2: Exquisitor's browser-based user interface. When hovering over an image, the user can label it as positive (bottom left), negative (bottom right), or seen (top right). Positive items (green column) and negative items (red column) are then used for updating the model.

Lifelogging is also steeped in history. In 1945, Vannevar Bush published an article in which he proposed the "Memex", which he described as "a device capable of storing all books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility" [1]. Despite the desire for such a device, the required technology did not exist and therefore Bush could only encourage future researchers to carry out this vision. The pioneering effort was the MyLifeBits project [5], where Gordon Bell attempted to digitize nearly every aspect of his life, creating the first lifelog. Recent years have seen the emergence of more devices capable of capturing lifelog data, such as miniature cameras, heart rate monitors, audio capturing devices, and GPSs, to name a few. Collecting all this information is the first step, of course, but as with the MyLifeBits project, the ability to process the data in real time at scale in a flexible way is still desired.

The LSC, now in its second year, is the first interactive challenge focusing on lifelog data. It derives its format from VideOlympics [22] and Video Browser Showdown (VBS) [20], inviting interactive retrieval systems to solve interactive tasks at premise. Six teams participated in LSC 2018. Some of these had previously participated in VBS, while others were new systems; overall the more developed systems had greater success [8]. The techniques of the different retrieval systems varied significantly, but features such as filters, similarity search and keyword search were a recurring theme [8]. On top of these, specific systems emphasized different interactions, such as virtual reality [3], sketch-based [14, 15] or visual concepts [16] to name a few. However, none of the LSC 2018 participants used a relevance feedback-based approach.

3 EXQUISITOR

Exquisitor is a user relevance feedback approach capable of handling large scale collections in real time [13]. It uses a Linear SVM classifier as the underlying model deployed to score items in a compressed feature space each interaction round. Furthermore it uses a high dimensional indexing approach based on extended Cluster

Pruning (eCP) [6]. The Exquisitor system used for LSC consists of three parts: (1) a web-based user interface for receiving and judging submissions; (2) an interactive learning server, which receives user judgments and produces a list of suggestions; and (3) an off-the-shelf web server which serves image thumbnails. In the following, we describe the first two parts of the system.

3.1 Exquisitor Interface

The current Exquisitor user interface, shown in Figure 2, is browser-based. It is largely a traditional interactive learning interface, in that users are asked to label positive and negative examples, which are then used to learn their preferences and determine the new round of suggestions. Due to the extreme efficiency of the scoring process, however, the interface itself initiates the request for new suggestions, either at regular intervals or when new examples have been produced.

3.2 Exquisitor Server

Exquisitor is developed to handle large-scale image collections, where each image is described with feature vector data from the visual and text modalities. The main components of the system are a) data representation and indexing, and b) the scoring process. We will briefly describe these in the following.

The high-dimensional feature vectors from the visual and text modality are independently compressed using an index-based compression method [24], where each feature vector is represented using the top 6 features of each modality and compressed into only three 64-bit integers. This results in an item only requiring 24 bytes of space per feature vector modality. The system has no need for decompression as it is capable of scoring the items directly in compressed space.

The compressed feature vectors are then indexed using the eCP high-dimensional indexing algorithm [7]. A set of R representative vectors is chosen from the collection and each vector is assigned to the closest representative, thus forming clusters in the compressed high-dimensional space. To facilitate retrieval, the clusters are recursively indexed, using the same method to select representatives of the representatives, to a chosen height L of the index.

In each interaction round, the Linear SVM model yields a classification hyperplane, which is used to form a farthest neighbor query to the cluster-based index. The goal is to yield k suggestions, which can be presented to the user. From each modality, b clusters are retrieved and their contents scanned to yield the r furthest neighbors from hyperplane. Using late modality fusion, these r candidates from each modality are then merged with a rank aggregation scheme to produce one ranked list, and the top k overall candidates returned. If further efficiency is required, multiple CPU cores can collaborate in producing the answer, by using w workers to process b/w clusters each.

Table 1 summarizes the initial parameter settings we have used for the LSC collection. Note that experiments with YFCC100M have shown that there is a tradeoff between latency and result quality. As more clusters are processed (higher b) both latency and result quality increase, but at some point result quality stops improving, and may even get worse with additional processing in some cases. We have yet to determine the optimal tradeoff between latency and result quality for the LSC data, but the collection is small enough to fully process in about 20 milliseconds per interaction round using only a single CPU core.

Table 1: Runtime parameters for Exquisitor with LSC data.

Parameter	Description	Default
Offline Indexing Parameters		
R	Number of representatives/clusters	417
L	Height of index tree	2
Runtime Scoring Parameters		
b	Clusters read from the index	16
r	Candidate items from each cluster	100
k	Number of new suggestions returned	25
w	Number of CPU cores used	1

4 DATASET PREPARATION

LSC 2019 provides a dataset consisting of lifelog data collected from a single user over the course of 27 days [9]. The dataset consists of 41.665 images with associated metadata and biometric data of the lifelogger. This section describes how the given data was processed into visual and textual feature vectors for use with Exquisitor.

4.1 Visual Data

In the LSC dataset, visual concepts (e.g., “computer”, “indoor”, and “wall”) have already been assigned to images with a certainty score ranging from 0 to 1. All in all, there are 548 unique concepts in the collection; the highest number of concepts found on a single image is 15. As described above, the 6 visual concepts with the highest certainty scores are retained in the compressed data representation, while the remaining concepts are ignored.

Note that not all images have visual concept data. A total of 986 images have apparently not successfully cleared the concept generation process and are not represented in the visual dataset at all. Additionally, 1,454 images had a “null” concept assignment, indicating that the feature extraction process yielded no concepts. As the data was processed sequentially, according to time and date, we have made the assumption that most images with missing visual concept data can be represented by the features of the previous successful image. Figure 3 shows one example where this assumption holds, but there are also examples where the previous image is far less similar.

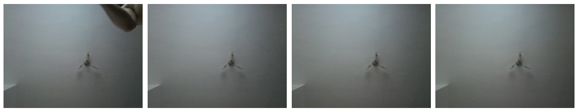
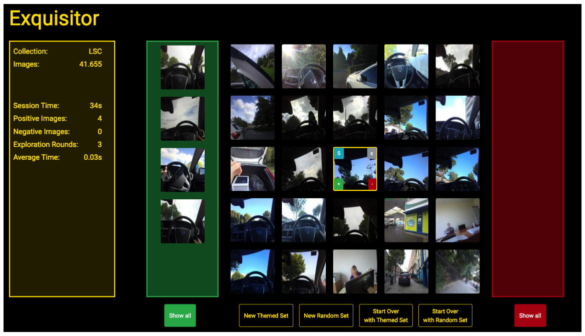


Figure 3: An example of consecutive images from the LSC dataset, where the first image has valid visual concepts while the following images have none.



Collection: LSC
Images: 41.655
Session Time: 34s
Positive Images: 4
Negative Images: 0
Exploration Rounds: 3
Average Time: 0.03s

Figure 4: Suggestions provided in the third round of interaction for an example AS task: find images of driving.

4.2 Textual Data

The text metadata consists of annotated descriptions for 23,788 images. Along with a general description of the activity in the image, the direct object with which the user was interacting, if any, is also given as text. These two fields were used to extract text feature scores, using a 100-topic LDA model trained on the English Wikipedia corpus using the gensim toolkit [18]. Note that 409 of the 986 items that had no visual features were found to have some textual features, leaving 577 images without any visual or text features.

4.3 Other Data

Additional metadata about the lifelog user were provided, such as location, heart-rate, food information, etc. For now, these have not been used to extract features, but this remains an option. Furthermore, this information could be used to combine filters with the relevance feedback process.

5 INITIAL EXPERIENCES

According to [23], the way a user initially interacts with a collection is by browsing through it. As further insight is gained about the collection and the task of the user becomes more clear, the user can start to narrow the scope until a result is achieved. The question then is whether this type of process is suitable for LSC tasks.

Based on the previous Lifelog Search Challenge, and other similar competitions such as Video Browser Showdown, the likely tasks for systems can be categorized into two groups: *Known Item Search* (KIS) and *Ad-hoc Search* (AS). In LSC 2018, the former was dominant. KIS tasks means that there is only one image (or a small set) that will satisfy the query, while AS tasks have more broad answers. In the following, we consider examples of AS and KIS tasks, and describe our initial experiences.

5.1 Example: Ad-hoc Search

As an example AS task, consider finding images where the user is driving. As can be seen in Figure 4, it took only 3 interaction rounds, starting from a random set of images, before the system became well aware of our intent and provided many relevant results. The overall process took a little over 30 seconds (left column), while producing suggestions took on average 30 milliseconds per interaction round.

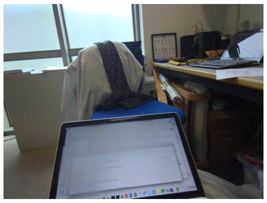


Figure 5: Random image chosen for the example KIS task.

5.2 Example: Known Item Search

We believe that a KIS task will be harder with a relevance feedback-based system, as finding a suitable Linear SVM model that separates the correct image from the collection will be hard. To test this, we have randomly chosen the image in Figure 5 as an example of a KIS task. Starting again from a random set of images, Exquisitor quickly identified that the information need included laptops or computers, but as the image is very similar to many other images containing laptops or computers, the correct image could not be found in 40 interaction rounds. Note that in LSC, each task generally has a set of images considered relevant, so this example KIS task is most likely significantly harder than LSC tasks, but it was nevertheless instructive, as summarized next.

5.3 Summary of Observations

So far, our work has been more focused on exploration than on identifying known items. While relevance feedback alone should be capable of narrowing the scope of exploration and eventually finding the correct items, some additional functionality appears necessary for the time-constrained LSC tasks. We have identified the following key issues to address before LSC 2019 starts:

(1) Using more modalities than only visual and text modalities is the first priority. Metadata, such as location, time, and day, could be used both to find candidates and influence their ranking, thus impacting the choice of suggestions. Furthermore, filters on metadata could be used to reduce the scope of exploration, thus allowing users to more quickly arrive at a correct answer.
(2) Currently, the initial set of images is chosen randomly. Using either a visual query or text query to prime the suggestions could be a good addition to the interface. Due to the underlying index structure, such queries can be easily implemented without changing the relevance feedback process.
(3) When looking for a known item, it must be possible to instruct the system that, while all of the suggestions shown are indeed relevant, none of them are *exactly* what is sought. In that case, the system should show further suggestions based on the same model.
(4) Currently, the interface only shows image thumbnails. Examining an image in more detail, along with its metadata, could help the user evaluate its relevance, and potentially also help choose which modalities to use or to adjust filters.

6 CONCLUSION

In this paper we have described the initial configuration of the Exquisitor system for our first participation in the Lifelog Search Challenge (LSC 2019). Exquisitor is a highly scalable interactive learning system, which relies on user relevance feedback to improve its model of the user's information need. What sets this system apart from related work is the scalability, which it owes to innovative feature selection, compression and indexing as well as the ability to train the interactive model and score multimedia items directly in the compressed space. As a consequence, the visual and text features for the LSC collection can be stored in less than 6MB of RAM and processed in about 30 milliseconds on average per interaction round on a modest laptop computer. We have described our initial experiences with using Exquisitor on lifelog data, and proposed a number of enhancements to the system for improved performance.

REFERENCES

[1] Vannevar Bush. 1945. As We May Think. *The Atlantic Monthly* 176, 1 (1945), 101–108.

[2] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. 1996. Active Learning with Statistical Models. *J. Artificial Intelligence Research* 4, 1 (1996), 129–145.

[3] Aaron Duane, Cathal Gurrin, and Wolfgang Huerst. 2018. Virtual Reality Lifelog Explorer: Lifelog Search Challenge at ACM ICMR 2018. In *Proceedings of the 2018 ACM Workshop on The Lifelog Search Challenge*. ACM, Yokohama, Japan, 20–23.

[4] Myron Flickner, Harpreet S. Sawhney, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, and Peter Yanker. 1995. Query by Image and Video Content: The QBIC System. *IEEE Computer* 28, 9 (1995), 23–32.

[5] Jim Gemmell, Gordon Bell, Roger Lueder, Steven Drucker, and Curtis Wong. 2002. MyLifeBits: Fulfilling the Memex Vision. In *Proc. ACM Multimedia*. ACM, Juan les Pins, France, 235–238.

[6] Gylfi Þór Guðmundsson, Laurent Amsaleg, and Björn Þór Jónsson. 2012. Impact of Storage Technology on the Efficiency of Cluster-based High-dimensional Index Creation. In *Proc. International Conference on Database Systems for Advanced Applications (DASFAA)*. Springer, Busan, South Korea, 53–64.

[7] Gylfi Þór Guðmundsson, Björn Þór Jónsson, and Laurent Amsaleg. 2010. A Large-scale Performance Study of Cluster-based High-dimensional Indexing. In *Proc. International Workshop on Very-large-scale Multimedia Corpus, Mining and Retrieval (VLS-MCMR)*. ACM, Firenze, Italy, 31–36.

[8] Cathal Gurrin, Klaus Schoeffmann, Hideo Joho, Andreas Leibetseder, Liting Zhou, Aaron Duane, Duc-Tien Dang-Nguyen, Michael Riegler, Luca Piras, Minh-Triet Tran, et al. 2019. Comparing Approaches to Interactive Lifelog Search at the Lifelog Search Challenge (LSC2018). *ITE Transactions on Media Technology and Applications* 7, 2 (2019), 46–59.

[9] Cathal Gurrin, Klaus Schoeffmann, Hideo Joho, Bernd Münzer, Rami Albatal, Frank Hopfgartner, Liting Zhou, and Duc-Tien Dang-Nguyen. 2019. A Test Collection for Interactive Lifelog Retrieval. In *Proc. International Conference on MultiMedia Modeling (MMM)*. Springer, Thessaloniki, Greece, 312–324.

[10] Cathal Gurrin, Alan F. Smeaton, and Aiden R. Doherty. 2014. LifeLogging: Personal Big Data. *Foundations and Trends in Information Retrieval* 8, 1 (2014), 1–125.

[11] Thomas S. Huang, Charlie K. Dagli, Shyamsundar Rajaram, Edward Y. Chang, Michael I. Mandel, Graham E. Poliner, and Daniel P. W. Ellis. 2008. Active Learning for Interactive Multimedia Retrieval. *Proc. IEEE* 96, 4 (2008), 648–667.

[12] Miriam W. Huijser and Jan C. van Gemert. 2017. Active Decision Boundary Annotation with Deep Generative Models. In *Proc. IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Venice, Italy, 5296–5305.

[13] Björn Þór Jónsson, Omar Shahbaz Khan, Hanna Ragnarsdóttir, Þórhildur Þorleiksdóttir, Jan Zahálka, Stevan Rudinac, Gylfi Þór Guðmundsson, Laurent Amsaleg, and Marcel Worring. 2019. Exquisitor: Interactive Learning at Large. arXiv:1904.08689.

[14] Jakub Lokoč, Gregor Kovalčík, Tomáš Souček, Jaroslav Moravec, and Přemysl Čech. 2019. VIRET: A Video Retrieval Tool for Interactive Known-Item Search. In *Proc. ACM International Conference on Multimedia Retrieval (ICMR)*. ACM, Ottawa, ON, Canada.

[15] Jakub Lokoč, Tomáš Souček, and Gregor Kovalčík. 2018. Using an interactive video retrieval tool for lifelog data. In *Proceedings of the 2018 ACM Workshop on The Lifelog Search Challenge*. ACM, Yokohama, Japan, 15–19.

[16] Bernd Münzer, Andreas Leibetseder, Sabrina Kletz, Manfred Jürgen Primus, and Klaus Schoeffmann. 2018. lifeXplore at the Lifelog Search Challenge 2018. In *Proceedings of the 2018 ACM Workshop on The Lifelog Search Challenge*. ACM, Yokohama, Japan, 3–8.

[17] Chris North. 2006. Toward Measuring Visualization Insight. *IEEE Computer Graphics and Applications* 26, 3 (2006), 6–9.

[18] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, Valletta, Malta, 45–50.

[19] Yong Rui, Thomas S. Huang, and Sharad Mehrotra. 1997. Content-Based Image Retrieval with Relevance Feedback in MARS. In *Proc. International Conference on Image Processing (ICIP)*. IEEE Computer Society, Santa Barbara, CA, USA, 815–818.

[20] Klaus Schoeffmann. 2014. A User-Centric Media Retrieval Competition: The Video Browser Showdown 2012-2014. *IEEE MultiMedia* 21, 4 (2014), 8–13.

[21] Klaus Schoeffmann, Werner Bailer, Cathal Gurrin, George Awad, and Jakub Lokoč. 2018. Interactive Video Search: Where is the User in the Age of Deep Learning?. In *Proc. ACM Multimedia*. ACM, Seoul, Republic of Korea, 2101–2103.

[22] Cees G. M. Snoek, Marcel Worring, Ork de Rooij, Koen E. A. van de Sande, Rong Yan, and Alexander G. Hauptmann. 2008. VideOlympics: Real-Time Evaluation of Multimedia Retrieval Systems. *IEEE MultiMedia* 15, 1 (2008), 86–91.

[23] Jan Zahálka and Marcel Worring. 2014. Towards Interactive, Intelligent, and Integrated Multimedia Analytics. In *Proc. of the IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE Computer Society, Paris, France, 3–12.

[24] Jan Zahálka, Stevan Rudinac, Björn Þór Jónsson, Dennis C. Koelma, and Marcel Worring. 2018. Blackthorn: Large-Scale Interactive Multimodal Learning. *IEEE Transactions on Multimedia* 20, 3 (2018), 687–698.

Exquisitor at the Lifelog Search Challenge 2020

Omar Shahbaz Khan
IT University of Copenhagen
Copenhagen, Denmark
omsh@itu.dk

Mathias Dybkjær Larsen
IT University of Copenhagen
Copenhagen, Denmark
mdyl@itu.dk

Liam Alex Sonto Poulsen
IT University of Copenhagen
Copenhagen, Denmark
liap@itu.dk

Björn Þór Jónsson
IT University of Copenhagen
Copenhagen, Denmark
bjorn@itu.dk

Jan Zahálka
Czech Technical University in Prague
Prague, Czech Republic
jan.zahalka@cvut.cz

Stevan Rudinac
University of Amsterdam
Amsterdam, Netherlands
s.rudinac@uva.nl

Dennis Koelma
University of Amsterdam
Amsterdam, Netherlands
d.c.koelma@uva.nl

Marcel Worring
University of Amsterdam
Amsterdam, Netherlands
m.worring@uva.nl

ABSTRACT

We present an enhanced version of Exquisitor, our interactive and scalable media exploration system. At its core, Exquisitor is an interactive learning system using relevance feedback on media items to build a model of the users' information need. Relying on efficient media representation and indexing, it facilitates real-time user interaction. The new features for the Lifelog Search Challenge 2020 include support for timeline browsing, search functionality for finding positive examples, and significant interface improvements. Participation in the Lifelog Search Challenge allows us to compare our paradigm, relying predominantly on interactive learning, with more traditional search-based multimedia retrieval systems.

CCS CONCEPTS

• **Information systems → Multimedia and multimodal retrieval**; **Multimedia databases**.

KEYWORDS

Lifelogging; Interactive learning; Exquisitor.

ACM Reference Format:
Omar Shahbaz Khan, Mathias Dybkjær Larsen, Liam Alex Sonto Poulsen, Björn Þór Jónsson, Jan Zahálka, Stevan Rudinac, Dennis Koelma, and Marcel Worring. 2020. Exquisitor at the Lifelog Search Challenge 2020. In *Proceedings of the Third Annual Workshop on the Lifelog Search Challenge (LSC '20), June 9, 2020, Dublin, Ireland.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3379172.3391718

1 INTRODUCTION

The Lifelog Search Challenge (LSC) is a live system-evaluation event, where researchers compare their systems based on their ability to help users quickly solve search-related tasks for a multimodal lifelog dataset. Each task in LSC is an independent query, to be solved in a few minutes, where a correct result is a single image returned from a set of relevant images. The query description is given gradually, as might be typical when a lifelog is used to find information and the user slowly remembers more details about the situation. The first two editions of LSC, held in 2018 [3, 4] and 2019 [5], have showcased a variety of multimedia retrieval systems aiming to search the lifelog with different approaches, ranging from traditional keyword search to novel virtual reality-based approaches (e.g., see [1, 9, 10, 12]).

We have recently developed Exquisitor, a highly scalable interactive learning system for general multimedia analytics applications [7]. When applied to LSC, the user is initially presented with a set of randomly selected images from the lifelog and asked to give feedback on (some of) the items about their relevance to the LSC task at hand. The feedback is used to build (and subsequently update) a classification model, which in turn is used to provide new suggestions; this iterative process continues as long as the user deems necessary. Figure 1 describes Exquisitor's interactive learning interface. A key feature that sets Exquisitor apart from other interactive learning approaches is its scalability: Exquisitor can retrieve suggestions from the LSC 2020 collection of 43K images in less than 50 milliseconds using a single CPU core, allowing to retrieve suggestions very rapidly following each user interaction.

Exquisitor participated in LSC 2019 [8], where it ranked sixth out of nine participants. The main lesson from LSC 2019 was that interactive learning is a viable approach, even in this heavily search-oriented competition setting. However, we also identified some shortcomings of the Exquisitor system itself that prevented solving some of the tasks. In this paper, we present the lessons learned from LSC 2019 and how we have improved the system for participation in LSC 2020. These improvements were partly implemented for participation in the Video Browser Showdown 2020 [6], where Exquisitor ranked fifth out of eleven participants.

LSC '20, June 9, 2020, Dublin, Ireland

ACM ISBN 978-1-4503-7136-0/20/06…$15.00
https://doi.org/10.1145/3379172.3391718

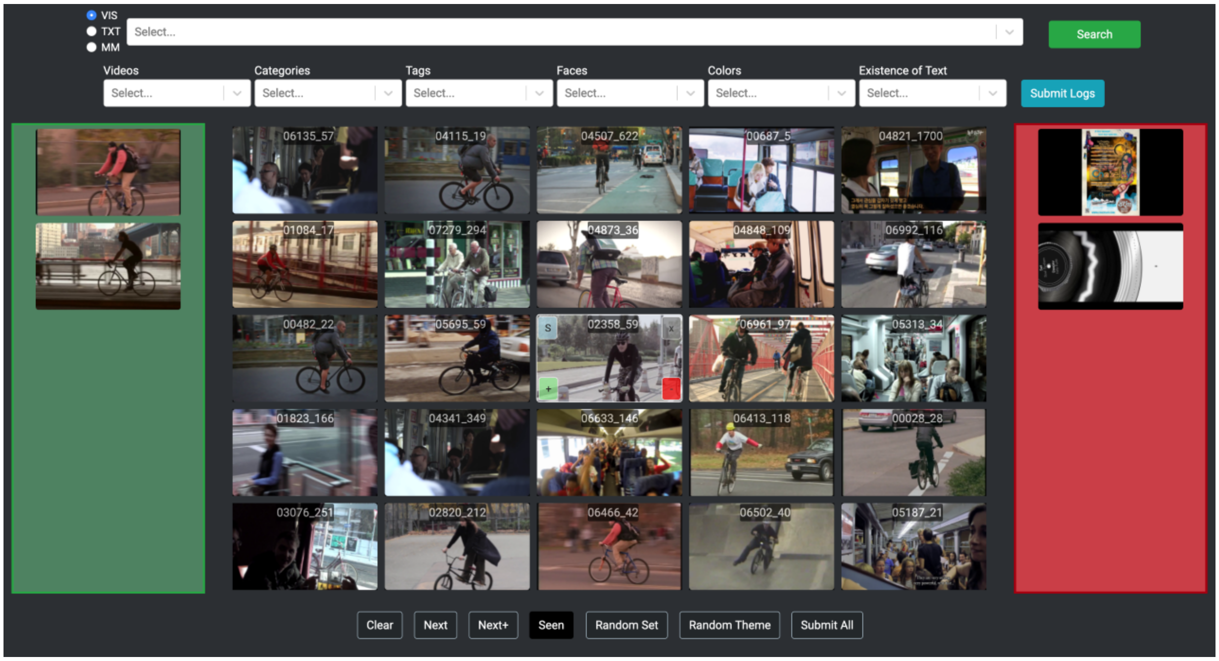


Figure 1: Exquisitor's interactive learning interface. Previously selected positive examples are shown on the left and negative examples on the right. The middle panel shows 25 suggestions based on the classification model built based on the user's feedback. By hovering over a thumbnail (see the middle thumbnail), users can select the image/video clip as a positive or negative example (bottom left/right corners), remove it from consideration (upper right corner) or submit as solution to the task (upper left corner). The top bars are for search and filtering, as described in the text.

The remainder of the paper is organized as follows. Section 2 briefly outlines the Exquisitor approach. Section 3 describes the lessons learned from participation in LSC 2019, and Section 4 reviews the changes made to Exquisitor based on those lessons.

2 EXQUISITOR

Exquisitor is a state-of-the-art multimodal interactive learning approach that combines efficient representation of data, a fast interactive classifier, and large-scale collection indexing [7]. The data representation for each multimodal item comprises state-of-the-art semantic visual concepts and text features. The semantic features are compressed per modality using an index-based compression method [16] that achieves over 99% compression rate whilst yielding a data representation that preserves the semantic information in the original data. The interactive classifier of choice, linear SVM, operates directly in the compressed space to greatly speed up the suggestion retrieval process. While more complex models, such as those based on CNN architectures, have achieved great successes in supervised learning settings, the performance of linear models for classification is still unparalleled in interactive learning due to their relatively good performance, explainability and the ability to scale to very large collections [7, 11, 13, 16].

To build an index suitable of scaling up to large scale datasets, Exquisitor builds on the extended Cluster Pruning (eCP) algorithm [2], which creates a hierarchical structure of the collection and enables efficient weaving of index utilization into the interactive learning pipeline. Instead of scoring all items in the collection with the classifier trained on user input, in each interaction round, Exquisitor first identifies the b clusters most relevant to the query, based on the SVM model, and then only scores items in those clusters, again using the SVM model to produce the suggestion candidates per modality. More specifically, the b clusters of each modality are divided into s segments, and a list of r candidates is produced from each segment. The final suggestions are then obtained by performing late modality fusion over the $s \times r$ candidates from each modality to produce the final k suggestions for the user.

By using a high-dimensional index, Exquisitor's suggestion retrieval relies not only on the scores provided by the interactive classifier, but also harnesses the collection's high-dimensional structure; our results indicate that this can indeed improve the quality of the suggestions at scale. In [7], large-scale, artificial actor-simulated experiments [15] with the ImageNet and YFCC100M collections show that with parameter settings of $b = 256, s = 16, r = 1,000$ and $k = 25$, Exquisitor significantly outperforms the state of the art in user relevance feedback.

3 LESSONS FROM LSC 2019

As outlined in the introduction, we believe that interactive learning as a concept performed quite well on the search-based tasks of LSC 2019. We found, however, that the system was missing some features that would have been useful for solving some of the tasks:

- *Model Bootstrapping.* Initially, the user is presented with a screen of 25 random images from the lifelog collection. Even for the relatively small LSC 2019 collection of about 43K images, this represents less than 0.1% of the collection. For some tasks there were few positive examples in the collection, so the odds of randomly finding positive examples was therefore very low. Some means of searching for positive examples is thus clearly needed.
- *Temporal Overview.* Several LSC tasks described a sequence of events leading up to the correct answer to the task, and sometimes these prior events were easier to identify than the eventual answer. Without any means to browse a timeline, finding these prior events offered limited value for solving the tasks.
- *General Interface Issues.* We found that the interactive learning interface itself had multiple problems, and was in particular difficult to use for novice users. This included basic issues such as too much unused space on the screen and too many mouse-clicks for common operations, as well as requiring complex interactions to apply filters to the relevance feedback process.
- *Metadata Integration.* Finally, at LSC 2019 we used only a subset of the available metadata. While the subset we used would have been sufficient to solve most of the tasks, integrating all available metadata is important for the ability to solve general analytics tasks.

We believe that these findings apply generally for any multimedia analytics application, as the problems encountered during LSC could be encountered in many situations where a combination of search and exploration is required.

4 NEW FEATURES FOR LSC 2020

In order to address the lessons described above, we have implemented the following changes to the Exquisitor system:

- *Model Bootstrapping.* We have implemented text-search functionality, using pylucene, over the metadata of the lifelog images, including the semantic concepts and their descriptions. Note, however, that the primary goal of the search functionality is not to find the answers to the tasks—although this may happen in some cases—but rather to identify positive example images, or even specific negative example images, that can be used to build the model of user intent.
- *Temporal Overview.* For the Video Browser Showdown, we implemented a video explorer to browse short scenes within the context of the videos, as shown in Figure 2. By considering each lifelog image as a thumbnail from a video (albeit, a video with a very low frame-rate), we adapt this functionality to support timeline browsing within the lifelog collection. We have also improved the timeline explorer implementation to provide flexible granularity of the lifelog timeline, thus providing better overview for the user.
- *General Interface Issues.* In order to improve usability, we have eliminated some functionality that was not used in practice (e.g., incrementally replacing images with new suggestions), streamlined several important operations (e.g., examining the collections of positive or negative examples), and improved screen usage significantly by eliminating unused background space.
- *Metadata Integration.* Finally, we are working to improve the use of images and metadata. We have applied state-of-the-art ResNeXt-101 visual concept detectors [14] to the lifelog images, impacting both the user relevance feedback process and text search. We have also improved the filtering process and are working to extend the range of metadata from the collection that is available to users. As an example, the ability to filter lifelog images based on geo-location could potentially be important for some LSC tasks.

As noted above, some of these enhancements have already been applied in our participation in the Video Browser Showdown 2020. With the additional changes made for LSC participation, we expect that the system will perform significantly better with LSC tasks.

5 CONCLUSION

Exquisitor is an efficient interactive learning system, which relies on user relevance feedback to build a model of the user's information need. While Exquisitor targets general multimedia analytics applications, the participation in the Lifelog Search Challenge (LSC) nevertheless allows comparison with more traditional search-based media retrieval systems. In this paper we have described the lessons learned from participation in LSC 2019 and the changes made to the Exquisitor system for our participation in LSC 2020.

ACKNOWLEDGMENTS

This work was supported by a PhD grant from the IT University of Copenhagen and by the European Regional Development Fund (project Robotics for Industry 4.0, CZ.02.1.01/0.0/0.0/15 003/0000470).

REFERENCES

[1] Aaron Duane, Cathal Gurrin, and Wolfgang Hürst. 2018. Virtual Reality Lifelog Explorer: Lifelog Search Challenge at ACM ICMR 2018. In *Proceedings of the ACM Workshop on Lifelog Search Challenge, LSC 2019*. ACM, Yokohama, Japan, 20–23.

[2] Gylfi Þór Guðmundsson, Björn Þór Jónsson, and Laurent Amsaleg. 2010. A Large-scale Performance Study of Cluster-based High-dimensional Indexing. In *Proc. International Workshop on Very-large-scale Multimedia Corpus, Mining and Retrieval (VLS-MCMR)*. ACM, Firenze, Italy, 31–36.

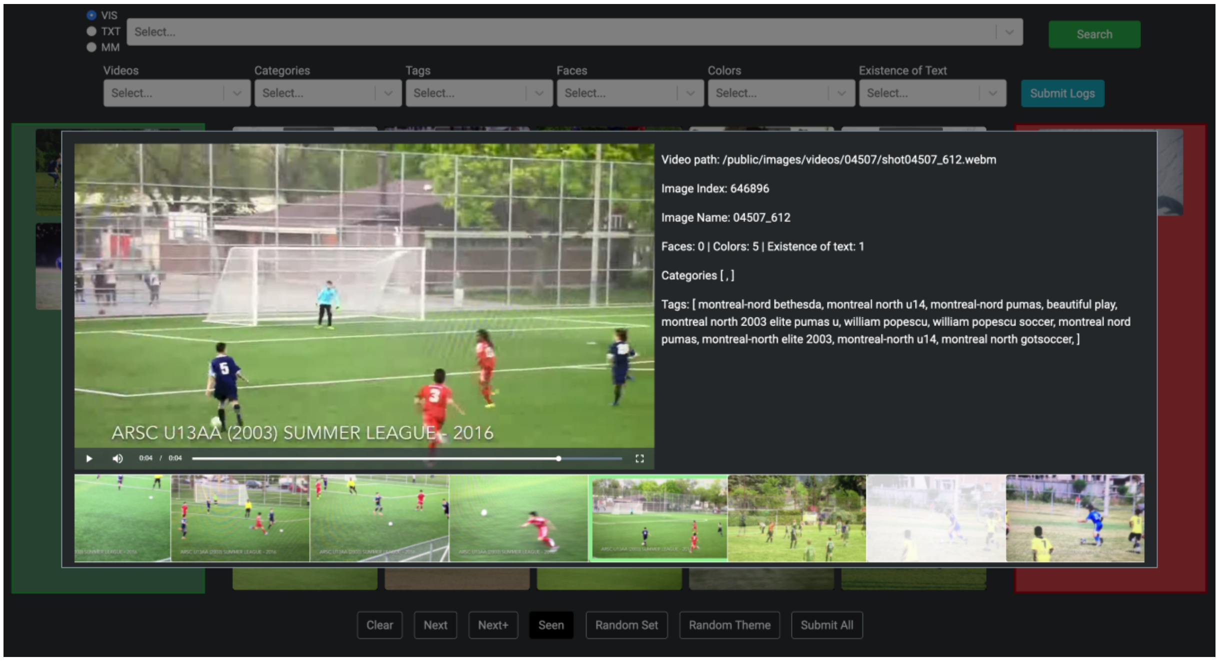
[3] Cathal Gurrin, Klaus Schoeffmann, Hideo Joho, Duc-Tien Dang-Nguyen, Michael Riegler, and Luca Piras (Eds.). 2018. *Proceedings of the ACM Workshop on Lifelog Search Challenge, LSC 2018*. ACM, Yokohama, Japan.

[4] Cathal Gurrin, Klaus Schoeffmann, Hideo Joho, Andreas Leibetseder, Liting Zhou, Aaron Duane, Duc-Tien Dang-Nguyen, Michael Riegler, Luca Piras, Minh-Triet Tran, et al. 2019. Comparing Approaches to Interactive Lifelog Search at the Lifelog Search Challenge (LSC2018). *ITE Transactions on Media Technology and Applications* 7, 2 (2019), 46–59.

[5] Cathal Gurrin, Klaus Schöffmann, Hideo Joho, Duc-Tien Dang-Nguyen, Michael Riegler, and Luca Piras (Eds.). 2019. *Proceedings of the ACM Workshop on Lifelog Search Challenge, LSC 2019*. ACM, Ottawa, ON, Canada.

[6] Björn Þór Jónsson, Omar Shahbaz Khan, Dennis C. Koelma, Stevan Rudinac, Marcel Worring, and Jan Zahálka. 2020. Exquisitor at the Video Browser Showdown 2020. In *Proceedings of the International Conference on MultiMedia Modeling (MMM)*. Springer, Daejeon, South Korea, 796–802.

[7] Omar Shahbaz Khan, Björn Þór Jónsson, Stevan Rudinac, Jan Zahálka, Hanna Ragnarsdóttir, Þórhildur Þorleiksdóttir, Gylfi Þór Guðmundsson, Laurent Amsaleg, and Marcel Worring. 2020. Interactive Learning for Multimedia at Large. In



Video path: /public/images/videos/04507/shot04507_612.webm

Image Index: 646896

Image Name: 04507_612

Faces: 0 | Colors: 5 | Existence of text: 1

Categories [,]

Tags: [montreal-nord bethesda, montreal north u14, montreal-nord pumas, beautiful play, montreal north 2003 elite pumas u, william popescu, william popescu soccer, montreal nord pumas, montreal-north elite 2003, montreal-north u14, montreal north gotsoccer,]

Figure 2: Exquisitor's interface for exploring media items (images or video clips) in a temporal context. The interface shows details of the metadata associated with the media item, and allows exploration of the temporal context.

Proceedings of the European Conference on Information Retrieval (ECIR). Springer, Lisboa, Portugal, 16.

[8] Omar Shahbaz Khan, Björn Þór Jónsson, Jan Zahálka, Stevan Rudinac, and Marcel Worring. 2019. Exquisitor at the Lifelog Search Challenge 2019. In *Proceedings of the ACM Workshop on Lifelog Search Challenge, LSC 2019.* ACM, Ottawa, ON, Canada, 7–11.

[9] Andreas Leibetseder, Bernd Münzer, Manfred Jürgen Primus, Sabrina Kletz, Klaus Schoeffmann, Fabian Berns, and Christian Beecks. 2019. lifeXplore at the Lifelog Search Challenge 2019. In *Proceedings of the ACM Workshop on Lifelog Search Challenge, LSC 2019.* ACM, Ottawa, ON, Canada, 13–17.

[10] Jakub Lokoč, Gregor Kovalčík, Tomáš Souček, Jaroslav Moravec, and Přemysl Čech. 2019. VIRET: A Video Retrieval Tool for Interactive Known-Item Search. In *Proc. ACM International Conference on Multimedia Retrieval (ICMR).* ACM, Ottawa, ON, Canada, 177–181.

[11] Ionuţ Mironică, Bogdan Ionescu, Jasper Uijlings, and Nicu Sebe. 2016. Fisher Kernel Temporal Variation-based Relevance Feedback for video retrieval. *Computer Vision and Image Understanding* 143 (2016), 38 – 51. https://doi.org/10.1016/j.cviu.2015.10.005 Inference and Learning of Graphical Models Theory and Applications in Computer Vision and Image Analysis.

[12] Luca Rossetto, Ralph Gasser, Silvan Heller, Mahnaz Amiri Parian, and Heiko Schuldt. 2019. Retrieval of Structured and Unstructured Data with vitrivr. In *Proceedings of the ACM Workshop on Lifelog Search Challenge, LSC 2019.* ACM, Ottawa, ON, Canada, 27–31.

[13] Sudheendra Vijayanarasimhan, Prateek Jain, and Kristen Grauman. 2014. Hashing Hyperplane Queries to Near Points with Applications to Large-Scale Active Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 2 (2014), 276–288.

[14] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated Residual Transformations for Deep Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE Computer Society, Honolulu, HI, USA, 5987–5995.

[15] Jan Zahálka, Stevan Rudinac, and Marcel Worring. 2015. Analytic Quality: Evaluation of Performance and Insight in Multimedia Collection Analysis. In *Proc. ACM Multimedia.* ACM, Brisbane, Australia, 231–240.

[16] Jan Zahálka, Stevan Rudinac, Björn Þór Jónsson, Dennis C. Koelma, and Marcel Worring. 2018. Blackthorn: Large-Scale Interactive Multimodal Learning. *IEEE Transactions on Multimedia* 20, 3 (2018), 687–698.

Exquisitor at the Video Browser Showdown 2020

Björn Þór Jónsson[1], Omar Shahbaz Khan[1], Dennis C. Koelma[2],
Stevan Rudinac[2], Marcel Worring[2], and Jan Zahálka[3]

[1] IT University of Copenhagen, Denmark
[2] University of Amsterdam, Netherlands
[3] Czech Technical University in Prague, Czech Republic

Abstract. When browsing large video collections, human-in-the-loop systems are essential. The system should understand the semantic information need of the user and interactively help formulate queries to satisfy that information need based on data-driven methods. Full synergy between the interacting user and the system can only be obtained when the system learns from the user interactions while providing immediate response. Doing so with dynamically changing information needs for large scale multimodal collections is a challenging task. To push the boundary of current methods, we propose to apply the state of the art in interactive multimodal learning to the complex multimodal information needs posed by the Video Browser Showdown (VBS). To that end we adapt the Exquisitor system, a highly scalable interactive learning system. Exquisitor combines semantic features extracted from visual content and text to suggest relevant media items to the user, based on user relevance feedback on previously suggested items. In this paper, we briefly describe the Exquisitor system, and its first incarnation as a VBS entrant.

Keywords: Interactive learning · Video browsing · Scalability.

1 Introduction

The Video Browser Showdown (VBS) is a series of annual live competitions, where researchers are asked to study and develop methods to solve search-related tasks for a benchmark video collection. The VBS tasks, which are independent queries of three different flavours, are unknown to the researchers, who must prepare their systems and data representations for any potential task. At competition time, users of all systems are then given a few minutes to solve the tasks. Furthermore, depending on the task, the query may be gradually refined by adding information as time passes, to simulate real users with imperfect memories. While the systems taking part in previous VBS editions employ a variety of advanced search and retrieval techniques, a common observation is that they are highly interactive, requiring users to review and refine results of queries, resulting in a highly interactive process. Interactive multimodal learning has been proposed as an interactive method capable of satisfying users with uncertain information needs [15]. Given the format of VBS, it is of significant academic interest to apply interactive multimodal learning to VBS.

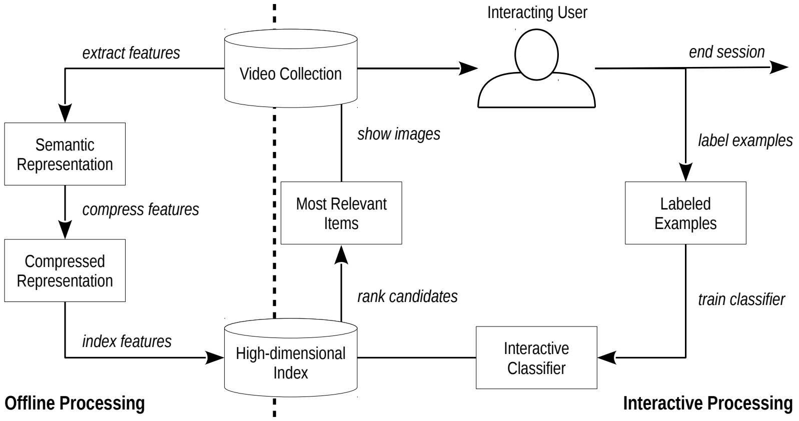


Fig. 1. Exquisitor's interactive learning pipeline. Initially, the video collection is processed to produce a compressed semantic representation, that is stored in a scalable high-dimensional index. In each round of the interactive learning process, the user is shown a set of potentially relevant videos. The user's judgments are then used to train a classifier, which in turn is used to retrieve a new set of videos to show to the user.

We have recently developed Exquisitor, a highly scalable interactive multimodal learning approach [5, 9]. Figure 1 illustrates the iterative feedback process employed by Exquisitor with video data. When a new task starts, the user is initially presented with a set of randomly selected video scenes from the collection and asked to give (positive or negative) feedback on (some of) the scenes. The feedback is used to build (and subsequently update) a classification model, which in turn is used to provide new suggestions; this iterative process continues as long as the user deems necessary. The Exquisitor system has been used to interactively explore the YFCC100M collection [9], and to compete in the Lifelog Search Challenge (LSC) 2019 [6], where it ranked 6th out of 9 competition entrants. A key feature that distinguishes Exquisitor from previous interactive learning systems is its scalability [5]; while the VBS video collection contains more than 1,000 hours of video, video suggestions can be retrieved in a fraction of a second in each interaction round. In this paper, we describe the adaptation of Exquisitor for participation in the Video Browser Showdown.

The remainder of the paper is organized as follows. In Sections 2 and 3, we briefly give background for interactive learning and the Video Browser Showdown, respectively. In Section 4, we then describe Exquisitor and its adaptation to VBS, before concluding in Section 5.

2 Interactive Learning

Interactive learning belongs to the family of human-in-the-loop learning approaches, eliciting data labels from the user and using that feedback to classify the otherwise unannotated data on the fly. In contrast to supervised learning, no labels are required prior to the analysis. Interactive learning commonly uses a lightweight, fast classifier that learns online as the user inputs her feedback.

The two main learning strategies in interactive learning are active learning and user relevance feedback. The objective of *active learning* is to create the best classifier by eliciting labels on data most informative to the classifier, which often translates to the data points the classifier is the least confident about or those closest to the decision boundary [1, 4]. Conversely, *user relevance feedback* aims to satisfy the user, presenting items for which the classification model is the most confident [11]. While this latter strategy may require more interactions to achieve the same final quality of the classification model, users may obtain their desired insights earlier [15].

The increasing drive towards interactivity, personalized user experience, and higher-level semantic understanding, combined with recent advances in related scientific disciplines [12, 15, 16], have motivated us to re-visit user relevance feedback with our Exquisitor approach [5, 9].

3 The Video Browser Showdown

Involving users in the evaluation of retrieval processes has long been a challenge [7, 12, 14]. The majority of multimedia and computer vision benchmark competitions are held offline, allowing scientists to devote both significant computational power and time, which has helped solve difficult closed-world problems. Over the last two decades, however, international interactive search benchmarking events have emerged, where systems and their users must solve unknown and complex tasks within a limited time frame. From its inception in 2001, the TRECVID benchmark initiative included an interactive search task [14]. The VideOlympics [13] then started in 2008 and ran for five years, introducing the concept of live interactive video search benchmarking. The Video Browser Showdown (VBS) has been running since 2012 [8], and is now the premier live event, where participants must explore and search a collection of 1,000 hours of video [10]. A recent event series is the Lifelog Search Challenge (LSC), where a collection of lifelog image data must be explored [3]. While VBS and LSC represent only subsets of multimedia analytics applications, participation is important as it allows comparison with related state-of-the-art interactive systems.

The tasks in VBS have three different flavours. Visual Known-Item-Search (KIS) tasks present a randomly selected video clip to competitors, who must then identify the correct clip in the collection and submit it to the VBS server. Textual KIS tasks present a gradually evolving text description, which again has a specific matching scene in the collection. Finally, Ad-hoc Video Search (AVS) tasks ask for scenes matching a description; in this task judges evaluate the relevance of answers as they are submitted to the VBS server. The VBS competition has an expert session, where the teams use their own systems to solve all types of tasks, and a novice session, where conference participants, who have never seen the system, are asked to solve visual KIS and AVS tasks.

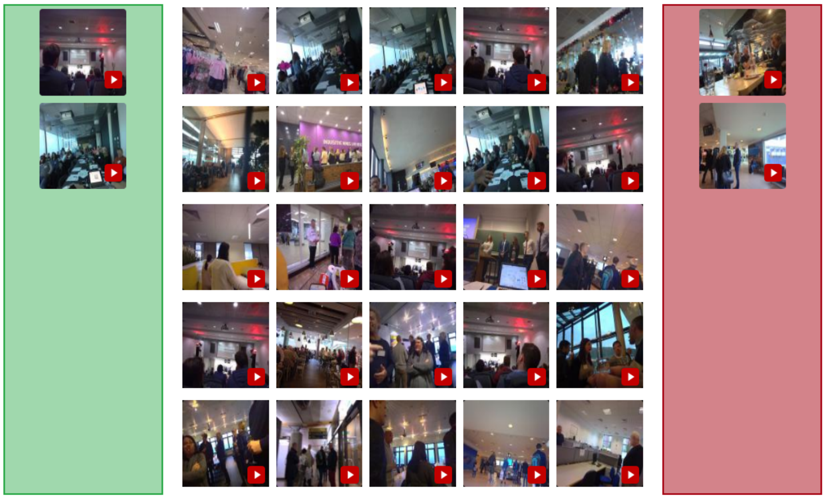


Fig. 2. Exquisitor's current user interface. The interface is browser-based and used primarily via mouse-based interaction. When hovering over a video, the user can choose to view the video in full, submit it to the VBS server, label it as a positive/negative example, or mark it as seen (using a 'next' button, the user can also mark all videos as seen and get a full screen of new videos). Positive (green column) and negative (red column) examples are immediately used to update the model.

4 Exquisitor

Exquisitor is a user relevance feedback approach capable of handling large scale collections in real time [5, 9]. The Exquisitor system used for VBS consists of three parts: (1) a web-based user interface for receiving and judging video suggestions; (2) an interactive learning server, which receives user judgments and produces a new round of suggestions; and (3) a web server which serves videos and video thumbnails. All three components run locally on the laptop of the VBS participants. In the following, we describe the first two parts of the system.

Exquisitor Interface: The current Exquisitor user interface is shown in Figure 2. In this initial incarnation, it is a pure interactive learning interface: the user is asked to label examples, which are subsequently used to learn the user's preference and suggest further examples. As the process to generate new suggestions is very efficient, however, new suggestions are retrieved each time the user identifies new positive or negative examples.

Exquisitor Server: Exquisitor has been developed to handle large-scale media collections, where each media item is described with feature vector data from both visual and text modalities. The main components of the server are a) data representation and indexing, and b) the scoring process, described briefly below.

Each of the (just over) million scenes in the VBS collection is represented by a high-dimensional concept feature vector extracted from a selected keyframe. The high-dimensional feature vectors are compressed using an index-based compression method [16], where each feature vector is represented using the top 6 features of the modality and compressed into only three 64-bit integers. The compressed feature vectors are then indexed using the eCP high-dimensional indexing algorithm [2]. A set of representative vectors is chosen from the collection and each vector is assigned to the closest representative, thus forming clusters in the compressed high-dimensional space. To facilitate retrieval, the cluster representatives are recursively indexed to form an approximate cluster-based index.

Exquisitor uses a Linear SVM classifier learned from user interactions to score items in the compressed feature space. In each interaction round, the Linear SVM model yields a classification hyperplane, which is used to form a farthest neighbor query to the cluster-based index. The goal is to yield $k = 25$ suggestions, which can be presented to the user. The clusters farthest from the SVM hyperplane are selected and their contents scanned to yield the k furthest neighbors.

Solving VBS Tasks: In KIS tasks, the aim of positive and negative examples is to create a model that is good enough to bring the correct answer to the screen. If the user is satisfied that all videos displayed are neither useful as positive/negative examples nor the answer to the task, the user can use the 'next' button to continue browsing the results, similar to the typical 'query and browse' approach of many current VBS entrants. A submitted result is considered as a positive example, regardless of whether it is the correct result or not; once the correct result has been submitted the task is complete. For AVS tasks the process is identical, except that all videos on screen can be submitted at once using a special button, and the process only ends once time has expired.

5 Conclusions

This paper has outlined the adaptation of the Exquisitor system to the Video Browser Showdown, both in terms of the data used to represent the video collection and the interface changes made for video browsing. As a new entrant in the competition, our primary goal is to learn from our participation in the competition, aiming to understand both how well the interactive learning approach suits the different competition tasks, and how we can improve our preliminary interface to be better suited to the competitive environment.

Acknowledgments: This work was supported by a PhD grant from the IT University of Copenhagen and by the European Regional Development Fund (project Robotics for Industry 4.0, CZ.02.1.01/0.0/0.0/15 003/0000470).

References

1. Cohn, D.A., Ghahramani, Z., Jordan, M.I.: Active learning with statistical models. J. Artificial Intelligence Research **4**(1), 129–145 (1996)
2. Guðmundsson, G.T., Jónsson, B.T., Amsaleg, L.: A large-scale performance study of cluster-based high-dimensional indexing. In: Proc. Int. Workshop on Very-large-scale Multimedia Corpus, Mining and Retrieval (VLS-MCM). Firenze, Italy (2010)
3. Gurrin, C., Schoeffmann, K., Joho, H., Dang-Nguyen, D., Riegler, M., Piras, L. (eds.): Proceedings of the 2018 ACM Workshop on The Lifelog Search Challenge, LSC@ICMR 2018. Yokohama, Japan (2018)
4. Huijser, M.W., van Gemert, J.C.: Active decision boundary annotation with deep generative models. In: Proc. IEEE ICCV. pp. 5296–5305. Venice, Italy (2017)
5. Jónsson, B.Þ., Khan, O.S., Ragnarsdóttir, H., Þorleiksdóttir, Þ., Zahálka, J., Rudinac, S., Guðmundsson, G.Þ., Amsaleg, L., Worring, M.: Exquisitor: Interactive learning at large. arXiv:1904.08689 (2019)
6. Khan, O.S., Jónsson, B.Þ., Zahálka, J., Rudinac, S., Worring, M.: Exquisitor at the Lifelog Search Challenge 2019. In: Proceedings of the ACM Workshop on Lifelog Search Challenge, LSC@ICMR. pp. 7–11. Ottawa, ON, Canada (2019)
7. Larson, M., Arora, P., Demarty, C., Riegler, M., Bischke, B., Dellandréa, E., Lux, M., Porter, A., Jones, G.J.F. (eds.): Working Notes Proceedings of the MediaEval 2018 Workshop, CEUR Workshop Proceedings, vol. 2283. CEUR-WS.org, Sophia Antipolis, France (2018)
8. Lokoč, J., Bailer, W., Schoeffmann, K., Münzer, B., Awad, G.: On influential trends in interactive video retrieval: Video Browser Showdown 2015-2017. IEEE Trans. Multimedia **20**(12), 3361–3376 (2018)
9. Ragnarsdóttir, H., Þorleiksdóttir, Þ., Khan, O.S., Jónsson, B.Þ., Guðmundsson, G.Þ., Zahálka, J., Rudinac, S., Amsaleg, L., Worring, M.: Exquisitor: Breaking the interaction barrier for exploration of 100 million images. In: Proceedings of the ACM Multimedia Conference. Nice, France (2019)
10. Rossetto, L., Schuldt, H., Awad, G., Butt, A.A.: V3C - A research video collection. In: Proc. MultiMedia Modeling (MMM). pp. 349–360. Thessaloniki, Greece (2019)
11. Rui, Y., Huang, T.S., Mehrotra, S.: Content-based image retrieval with relevance feedback in MARS. In: Proc. ICIP. pp. 815–818. Santa Barbara, CA, USA (1997)
12. Schoeffmann, K., Bailer, W., Gurrin, C., Awad, G., Lokoč, J.: Interactive video search: Where is the user in the age of deep learning? In: Proc. ACM Multimedia. pp. 2101–2103. Seoul, Republic of Korea (2018)
13. Snoek, C.G.M., Worring, M., de Rooij, O., van de Sande, K.E.A., Yan, R., Hauptmann, A.G.: Videolympics: Real-time evaluation of multimedia retrieval systems. IEEE MultiMedia **15**(1), 86–91 (2008)
14. Thornley, C., Johnson, A.C., Smeaton, A.F., Lee, H.: The scholarly impact of TRECVID (2003-2009). Journal of the American Society for Information Science and Technology (JASIST) **62**(4), 613–627 (2011)
15. Zahálka, J., Worring, M.: Towards interactive, intelligent, and integrated multimedia analytics. In: Proc. of the IEEE Conference on Visual Analytics Science and Technology (VAST). pp. 3–12. Paris, France (2014)
16. Zahálka, J., Rudinac, S., Jónsson, B.T., Koelma, D.C., Worring, M.: Blackthorn: Large-scale interactive multimodal learning. IEEE Transactions on Multimedia **20**(3), 687–698 (2018)

Exquisitor at the Video Browser Showdown 2021: Relationships Between Semantic Classifiers

Omar Shahbaz Khan[1], Björn Þór Jónsson[1], Mathias Larsen[1], Liam Poulsen[1], Dennis C. Koelma[2], Stevan Rudinac[2], Marcel Worring[2], and Jan Zahálka[3]

[1] IT University of Copenhagen, Denmark
[2] University of Amsterdam, Netherlands
[3] Czech Technical University in Prague, Czech Republic

Abstract. Exquisitor is a scalable media exploration system based on interactive learning, which first took part in VBS in 2020. This paper presents an extension to Exquisitor, which supports operations on semantic classifiers to solve VBS tasks with temporal constraints. We outline the approach and present preliminary results, which indicate the potential of the approach.

Keywords: Interactive learning · Video browsing · Temporal relations.

1 Introduction

The Video Browser Showdown (VBS), now in its 10th anniversary edition, has emerged as an important vehicle for the evolution of the multimedia field [5]. During VBS, researchers are given a series of never-before-seen task descriptions, based on a collection of 7,475 video clips [9], and asked to interactively retrieve either one specific video segment or multiple relevant segments, depending on the task type. VBS allows researchers working on media exploration and search tools to apply their techniques in a realistic setting and better understand the pros and cons of both the underlying techniques and the interfaces. The lessons learned during the competition can then inspire new methods and further research. In addition, the competitive setting makes for an exciting event where the ranking of systems can also give hints to their usability and applicability.

Exquisitor, a prototype media exploration system based on interactive learning, took part for the first time in VBS 2020, where it placed 5th out of 11 systems [2]. The goal of Exquisitor, as applied to VBS, is to build a semantic classifier for the information need represented in each task, and use that classifier—along with metadata filters and a video timeline explorer—to solve the task. Exquisitor uses the video segmentation supplied with the VBS collection and represents each video segment independently by semantic features derived from its keyframe. When building the semantic classifier, Exquisitor suggests keyframes to the user and asks for feedback on those suggestions. Once the user spots a potentially relevant keyframe, the video explorer can then be used to explore the actual content and internal structure of the full video clip.

For many VBS tasks, the task description applies to more than one video segment, often focusing on different semantic concepts in different segments, and sometimes providing an explicit temporal relationship. Unsurprisingly, therefore, all the strongest VBS competitors provide temporal queries as a major technique [4, 6, 10, 7]. Since video segmentation tends to split the video by semantic concepts, a classifier built to find one segment may not find the other, and the system should provide support to utilise the relationship between concepts in video segments.

In this paper, we present a new version of Exquisitor, where the major extension is the support for utilising relationships between semantic classifiers. While each semantic classifier is developed in the same manner as before, using independent video segments, the results of two semantic classifiers can now be combined in various ways, with an optional temporal relationship specification. In this paper we briefly outline the method and interface for combining two semantic models and show how two models combined could be used to solve two VBS 2020 tasks, one of which the team failed to solve during the competition. We will present and evaluate the methods in more detail in a later publication.

2 Exquisitor

Exquisitor is a user relevance feedback approach capable of handling large scale collections in real time [3, 8]. The Exquisitor system used for VBS consists of three parts: (1) a web-based user interface for receiving and judging video suggestions; (2) an interactive learning server, which receives user judgments and produces a new round of suggestions; and (3) a web server which serves videos and video thumbnails. Due to the computational efficiency of the system, all three components can run locally on a laptop.

Exquisitor Server: Exquisitor is fueled by a semantic model that combines interactive multimodal learning with cluster-based indexing. Each keyframe in each modality is represented by an efficient representation containing the most important semantic features, compressed using an index-based method [11]. This representation is further clustered using a cluster-based indexing approach [1]. When building a semantic classifier C, a linear SVM classifier is iteratively refined based on user interactions (positive and negative examples). In each round of interaction, the resulting separating hyperplane forms k-farthest neighbour queries posed to the cluster-based indexes. Finally, late fusion is performed on the retrieved results, to produce the 25 top-ranked results to suggest to the user.

Exquisitor Interface: The interface for building classifiers is shown in Figure 1. By hovering over a keyframe, the user can choose to view the video, submit it to the VBS server, label it as a positive/negative example, or mark it as seen. Using the 'next' button, the user can also mark all videos as seen and get a full screen of new videos based on the current semantic classifier. Positive (green column) and negative (red column) examples are immediately used to update the model.

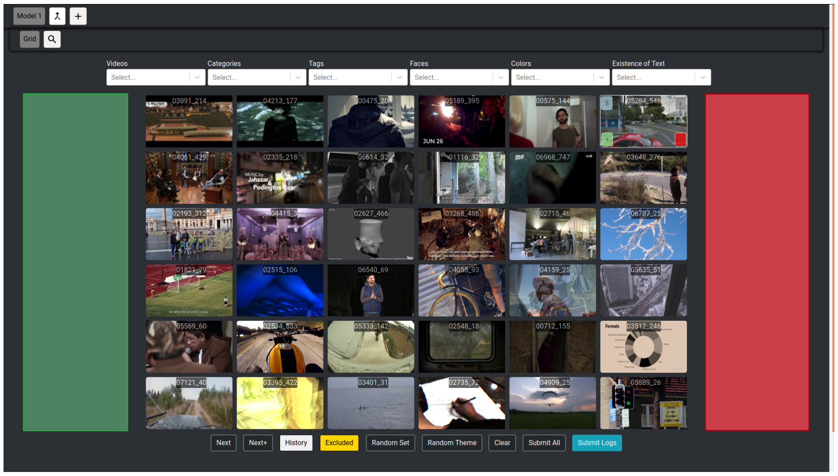


Fig. 1. Exquisitor's interface for building semantic classifiers. See text for details.

Interactive Learning and VBS: The tasks in VBS have three different flavours: Textual Known-Item-Search (KIS) tasks present a gradually evolving text description matching a short video segment; Visual KIS tasks show the video clip sought; and Ad-hoc Video Search (AVS) tasks ask for all segments matching a description. In these tasks, the aim of interactive learning is to create a classifier that is good enough to bring the correct answer(s) to the screen. For KIS tasks, a submitted result is considered as a positive example; once the correct result has been submitted the task is complete. For AVS tasks the process is identical, except that all videos on screen can be submitted at once using a special button, and the process only ends once time has expired.

3 Operations on Semantic Classifier Rankings

To ground the presentation, consider the two textual KIS tasks in Table 1, both of which have a temporal component. Task T_1 was solved by 6 teams during VBS 2020, and was generally considered a difficult task. There are many videos with bridesmaids and brides and grooms, respectively, but in this particular video they do not co-occur in a keyframe during the segment that was considered a solution to the task, and hence we failed to solve this task. Task T_6, on the other hand, was the only text-based task solved by all teams. The Exquisitor team solved it efficiently during the competition by building a classifier for elevators, since (a) the elevator and the bike co-exist in the same keyframe and (b) elevators are rare, so the keyframe is quickly suggested for inspection. Note, however, that since there are many examples of bikes in the collection, but most of them outdoors, building a classifier for bikes is not a productive method to solve T_6.

Table 1. Two example textual KIS tasks from VBS 2020.

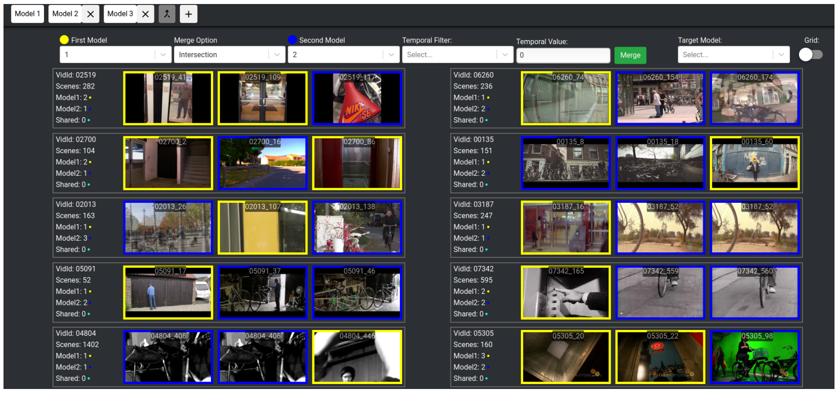
Task	Description
T_1	Seven bridesmaids in turquoise dresses walking down a street, and three still images of the bride and couple. The bridesmaids walk on the sidewalk towards the camera. The photos of the couple and bride are taken in a park.
T_6	Red elevator doors opening, a bike leans inside, doors closing and reopening, bike is gone. Zoom-in on bike, zoom-out from empty elevator. The bike is silver, the text 'ATOMZ' is visible.

Classifier Ranking Operations: The rankings obtained by two semantic classifiers, C_1 and C_2, can be combined with a keyframe relationship operation, C_1 *op* C_2, where $op \in \{\cap, \cup, \setminus, \dot{-}\}$. Furthermore, a temporal constraint can optionally be added, which requires either a maximum distance between keyframes (*within <frames>*) or a minimum distance (*after <frames>*). The result of the classifier ranking operation is a list of videos satisfying both the relationship constraint and optional temporal constraint. Each video is represented by a list of keyframes, annotated by the classifier(s) they appear in, and the videos are ranked by an average score based on the accumulated rank of their scenes from each classifier and the total number of scenes.

As an example, consider solving task T_6 by intersection of rankings produced by semantic classifiers for bikes and elevators. A video would be returned as an answer only if both classifiers return a scene from that video. Since the task description indicates that the two elements should be close to each other, a temporal constraint of *within* 1, for example, would avoid videos where bikes and elevators are far apart.

User Interface: Figure 2 shows the interface for classifier ranking operations. As the figure shows, the result of the merge is a list of the 10 top-ranked videos, where each video is represented by three colour-coded keyframes. Yellow keyframes are from C_1 and blue from C_2, while keyframes appearing in both classifiers are shown as green. The interface shows the highest ranked frame of each colour; if no keyframe appears in both classifiers, the third frame is the second highest frame from one classifier. Additionally, summary information on the number of keyframes in the video and classifiers is shown to the left of the keyframes.

Evaluation: To evaluate the usefulness of classifier ranking operations, we attempted to solve the two tasks of Table 1, both by building a single classifier and by building two classifiers and intersecting their rankings. These experiments were carried out in a calm setting, with no time limit, unlike the competitive environment of VBS. Furthermore, for this evaluation, the entire task text was considered. To estimate the user workload, we counted the number of interactions with the system, where an interaction is any action taken by the user, such



First Model: 1
Merge Option: Intersection
Second Model: 2
Temporal Filter: Select...
Temporal Value: 0
Target Model: Select...
Grid:

VidId: 02519
Scenes: 282
Model1: 2
Model2: 1
Shared: 0

VidId: 02700
Scenes: 104
Model1: 2
Model2: 1
Shared: 0

VidId: 02013
Scenes: 163
Model1: 1
Model2: 3
Shared: 0

VidId: 05091
Scenes: 52
Model1: 1
Model2: 2
Shared: 0

VidId: 04804
Scenes: 1402
Model1: 1
Model2: 1
Shared: 0

VidId: 06260
Scenes: 236
Model1: 1
Model2: 2
Shared: 0

VidId: 00135
Scenes: 151
Model1: 1
Model2: 2
Shared: 0

VidId: 03187
Scenes: 247
Model1: 1
Model2: 1
Shared: 0

VidId: 07342
Scenes: 595
Model1: 2
Model2: 2
Shared: 0

VidId: 05305
Scenes: 160
Model1: 3
Model2: 2
Shared: 0

Fig. 2. Exquisitor's interface for semantic classifier operations. See text for details.

as labelling a keyframe as a positive example or changing to a different interface component. We chose to stop after around 75 interactions; once we reached this limit, we considered the task to be unsolved.

Table 2 summarises the results for the two tasks. Consider first T_1, a difficult task which was not solved by Exquisitor during the competition. As the table shows, a simple intersection of the results produced by two classifiers could solve the task. Now consider task T_6, which was significantly easier. Table 2 shows that a single classifier on 'elevator' is the fastest approach to solve this task, due to the composition of the collection; this was fortunately the approach taken during the competition. Had we chosen to focus on 'bike' instead, however, the results suggest we would have failed to solve the task. Building rough classifiers for each concept and intersecting their rankings, however, is also an efficient method to solve the task; since the order in which the models are built does not matter the method is robust.

Table 2. Effectiveness experiment results

Task	Models	Interactions	Solved
T_1	'bridesmaid'	76	No
	'bride'	78	No
	'bridesmaid' ∩ 'bride'	60	Yes
T_6	'elevator'	8	Yes
	'bike'	75	No
	'elevator' ∩ 'bike'	15	Yes

4 Conclusions

We have outlined an extension to the Exquisitor system, supporting operations on semantic classifiers to solve VBS tasks with temporal constraints. Our preliminary results indicate that this new approach has significant potential, and we look forward to testing the approach in the competitive setting.

Acknowledgments: This work was supported by a PhD grant from the IT University of Copenhagen and by the European Regional Development Fund (project Robotics for Industry 4.0, CZ.02.1.01/0.0/0.0/15 003/0000470).

References

1. Guðmundsson, G.Þ., Jónsson, B.Þ., Amsaleg, L.: A large-scale performance study of cluster-based high-dimensional indexing. In: Proc. Int. Workshop on Very-large-scale Multimedia Corpus, Mining and Retrieval (VLS-MCM). Firenze, Italy (2010)
2. Jónsson, B.Þ., Khan, O.S., Koelma, D., Rudinac, S., Worring, M., Zahálka, J.: Exquisitor at the Video Browser Showdown 2020. In: Proc. MultiMedia Modeling (MMM). Daejeon, South Korea (2020)
3. Khan, O.S., Jónsson, B.Þ., Rudinac, S., Zahálka, J., Ragnarsdóttir, H., Þorleiksdóttir, Þ., Guðmundsson, G.Þ., Amsaleg, L., Worring, M.: Interactive learning for multimedia at large. In: Proc. European Conference on IR Research (ECIR) (2020)
4. Kratochvíl, M., Veselý, P., Mejzlík, F., Lokoč, J.: SOM-Hunter: Video browsing with relevance-to-SOM feedback loop. In: Proc. MultiMedia Modeling (MMM). Daejeon, South Korea (2020)
5. Lokoč, J., Kovalčík, G., Müncher, B., Schöffmann, K., Bailer, W., Gasser, R., Vrochidis, S., Nguyen, P.A., Rujikietgumjorn, S., Barthel, K.U.: Interactive search or sequential browsing? A detailed analysis of the Video Browser Showdown 2018. ACM TOMM **15**(1) (2019)
6. Lokoč, J., Kovalčík, G., Souček, T.: VIRET at Video Browser Showdown 2020. In: Proc. MultiMedia Modeling (MMM). Daejeon, South Korea (2020)
7. Nguyen, P.A., Wu, J., Ngo, C.W., Francis, D., Huet, B.: VIREO @ Video Browser Showdown 2020. In: Proc. MultiMedia Modeling (MMM). Daejeon, South Korea (2020)
8. Ragnarsdóttir, H., Þorleiksdóttir, Þ., Khan, O.S., Jónsson, B.Þ., Guðmundsson, G.Þ., Zahálka, J., Rudinac, S., Amsaleg, L., Worring, M.: Exquisitor: Breaking the interaction barrier for exploration of 100 million images. In: Proc. ACM Multimedia. Nice, France (2019)
9. Rossetto, L., Schuldt, H., Awad, G., Butt, A.A.: V3C - a research video collection. In: Proc. MultiMedia Modeling (MMM). Thessaloniki, Greece (2019)
10. Sauter, L., Parian, M.A., Gasser, R., Heller, S., Rossetto, L., Schuldt, H.: Combining boolean and multimedia retrieval in vitrivr for large-scale video search. In: Proc. MultiMedia Modeling (MMM). Daejeon, South Korea (2020)
11. Zahálka, J., Rudinac, S., Jónsson, B.Þ., Koelma, D.C., Worring, M.: Blackthorn: Large-scale interactive multimodal learning. IEEE TMM **20**(3) (2018)

Chapter 5

Conclusion

The aim of this thesis has been to design an interactive learning approach for multimedia collections, which can be used to determine the benefits and shortcomings of such approaches, when dealing with large collections and tasks that focus on exploration and search. Prior large-scale interactive learning approaches have shown adequate performance, but they either lean too much towards the machine's representation of multimedia items, making it difficult for a human to assess, or heavily rely on available computing resources to become a scalable approach. As interactive learning is a human-in-the-loop approach, it is important that the machine is transparent with the human to make the most out of the interactions. Furthermore, it is important to remember that everyone does not have the same availability of computing resources. Therefore, a scalable approach should not have a heavy reliance on computing resources, to ensure a larger appeal. Large-scale content based retrieval approaches rely on high-dimensional indexing which heavily focuses on solving search-oriented tasks, and have rarely been used for interactive learning with an emphasis on both exploration and search. For high-dimensional indexing to be used with interactive learning, a set of requirements are proposed in this thesis, which ensure that it is responsive and scalable. Furthermore, the interactive learning approach needs to be flexible to adapt towards shifting tasks, and accurate at scale. In this thesis the Exquisitor approach is proposed, which combines interactive learning with high-dimensional indexing following these requirements. To achieve these requirements, the interactive learning classifier and high-dimensional index have to be aligned, when it comes to representation and what a relevant item is. Otherwise, the process is not transparent and it becomes cumbersome to determine where potential errors arise. Similarly, the high-dimensional index has the property of being approximate, which is useful for controlling the time and quality, but there needs to be a way for the approach to circumvent this property when the user is only interested in small groups of items across the collection. Exquisitor uses a compressed representation for the multimedia items, which the high-dimensional index stores and uses as is. The interactive classifier of SVM is used to train a hyperplane, to which the farthest items from the plane in the positive direction are seen as relevant. The high-dimensional index has been modified to perform k-farthest neighbor queries to a plane, to find relevant items. Lastly, to circumvent the approximation property, incremental retrieval and query optimisation policies are present in Exquisitor.

During the work on Exquisitor it became apparent that automated evaluation measures do not consider the behaviors of different human users when interacting with such an approach. To this end, we have shown that how a user interacts with a system can have a significant impact on the outcome of a session for a specific task. This is shown through new evaluation protocols, but also observed through live showcases of Exquisitor as a client application. At these showings, it is evident that a pure interactive learning approach requires additional features, such as procuring initial positive examples, making corrections by replacing or removing already labeled items, and having multiple models to deal with temporal tasks. These features are present in the Exquisitor client stemming from the work in this thesis. Furthermore, the representations used for the multimedia items in the interactive learning process have primarily been visual semantic concepts, which are suitable for image collections, but for videos additional modality representations can be involved. While additional representations are possible to extract, through our work we have shown that it is important to keep in mind whether or not they are useful for a given task and collection. Furthermore, it may be better to discard weaker modalities or set preferences on the stronger modalities, when it is clear that a weak modality has little value for a task.

Exquisitor has shown that a truly scalable interactive learning approach is capable of exploring and searching large collections and acquiring desired knowledge. Furthermore, it is able to adapt towards new areas in case the nature of the task changes as knowledge is gained throughout a session. Thus, making Exquisitor responsive, accurate, flexible and scalable, using significantly less computing resources than its predecessors. Following this work, there are multiple research avenues to take up with regards to the foundation of Exquisitor, automated evaluation protocols, and better utilisation of the newfound available resources.

With continuous advances in deep learning for representations of image and video content, it is important to check whether these can replace current representations, or be added as additional representations. Cross-modal embeddings combine the information from multiple modalities and learn a new multi-modal representation, which have been shown to outperform single modality representations in most cases [16, 50, 88]. Contrastive learning is another approach that combines the information from text and visual content through self supervised learning, and has shown immense improvements for generalized image and video captioning, when using a large number of training samples [8, 81, 94]. These representations are often used for retrieval focusing on search, hence it will be interesting to see whether they are useful for interactive learning. Another representation is based on hypergraphs, where the relations between the various representations connected to multimedia items are used to create a hypergraph from which a singular representation is learned [4]. In this case it will be interesting to see how well the hypergraph representation compares to multiple representations.

When it comes to evaluating interactive learning approaches, the automated evaluation measures lack the influence of a real human user. While the new evaluation protocols presented in this thesis attempt to improve the artificial users, by introducing labeling and filtering strategies, there are still many details of a human user's behavior that are not represented. An approach to create more intelligent artificial users, is to use reinforcement learning to generate different users [77]. This requires an initial data collection phase from multiple user sessions, to capture the various behaviors of real human users. By analysing the session data, different policies, rewards, and states, can be defined from the different actions users make, which can be used by the reinforcement learning agents. As there are many actions a user can take in the Exquisitor client application, the most frequently used actions that focus on the interactive learning process should be considered.

Finally, Exquisitor reducing computing resources for performing interactive learning, opens up for the opportunity of utilising the additional resources to improve the overall experience of an interactive session. This may include multiple different classifiers for the interactive learning approach such as a linear SVM and a Self-Organising-Map. Furthermore, resources can be used to facilitate two completely different sessions, from the same or another user, which can be used to combine the results of both, similar to how the Exquisitor client handles temporal queries. With regards to the client application, resources can be used to introduce new retrieval methods, to determine potential filters from a given query, or to improve the interactive learning by adding methods for finding better examples. Another aspect of this work is that we have shown that even with weaker deep learning model representations, Exquisitor is capable of getting relevant items. Thus, in case of time dependent tasks, where a multimedia collection needs to be analysed, preprocessing a collection with multiple models, both stronger and weaker models, is a possibility, where the weaker model's representations are used to access and explore the collection quickly, and when the stronger model's representations are ready, they can either replace or work with the representations of the other model.

Ultimately, the work stemming from this thesis is an interactive learning approach that allow users to spend more time interacting with their collections in continuous sessions that adequately shifts to their needs. By significantly reducing resource requirements, there is now a way to interact with a large collection without having to rely on a cloud platform or specialized hardware. Thus, people or parties with collections they wish to analyse that contain private or sensitive data, now have a solution for this. There is still the notion of performing feature extraction which may demand specialized hardware or more resources, but this is a one-time operation. Lastly, Exquisitor shows that everyone has the ability to interact with such large collections in a manner that is not reliant on us formulating understandable terms for a machine and more importantly, in a global society, is independent of language.

Bibliography

[1] Charu C. Aggarwal, Xiangnan Kong, Quanquan Gu, Jiawei Han, and Philip S. Yu. Active learning: A survey. In *Data Classification*, pages 571–605. CRC Press, 2014.

[2] Giuseppe Amato, Paolo Bolettieri, Fabio Carrara, Franca Debole, Fabrizio Falchi, Claudio Gennaro, Lucia Vadicamo, and Claudio Vairo. The VISIONE Video Search System: Exploiting Off-the-Shelf Text Search Engines for Large-Scale Video Retrieval. *Journal of Imaging*, 7(5):76, 2021.

[3] Alexandr Andoni and Piotr Indyk. Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 459–468. IEEE, 2006.

[4] Devanshu Arya, Stevan Rudinac, and Marcel Worring. HyperLearn: A Distributed Approach for Representation Learning in Datasets With Many Modalities. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM'19, page 2245–2253. ACM, 2019.

[5] Elham Bagherian and Rahmita Wirza O.K. Rahmat. Facial feature extraction for face recognition: A review. In *2008 International Symposium on Information Technology*, volume 2, pages 1–9. IEEE, 2008.

[6] Alexandra M Bagi, Kim Ida Schild, Omar Shahbaz Khan, Jan Zahálka, and Björn Þór Jónsson. XQM: Interactive Learning on Mobile Phones. In *Proceedings of the 27th International Conference on Multimedia Modeling*, MMM'21, pages 281–293. Springer, 2021.

[7] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443, 2018.

[8] Manuele Barraco, Marcella Cornia, Silvia Cascianelli, Lorenzo Baraldi, and Rita Cucchiara. The Unreasonable Effectiveness of CLIP Features for Image Captioning: An Experimental Analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4662–4670, 2022.

[9] Ronen Basri, Tal Hassner, and Lihi Zelnik-Manor. Approximate Nearest Subspace Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):266–278, 2011.

[10] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[11] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When Is "Nearest Neighbor" Meaningful? In *Database Theory — ICDT'99*, pages 217–235. Springer, 1999.

[12] Christian Böhm. A Cost Model for Query Processing in High Dimensional Data Spaces. *ACM Transactions on Database Systems*, 25(2):129–178, 2000.

[13] Christian Böhm, Stefan Berchtold, and Daniel A Keim. Searching in High-Dimensional Spaces: Index Structures for Improving the Performance of Multimedia Databases. *ACM Computing Surveys*, 33(3):322–373, 2001.

[14] Christian Böhm and Hans-Peter Kriegel. Dynamically Optimizing High-Dimensional Index Structures. In *Proceedings of the 7th International Conference on Extending Database Technology*, EDBT 2000, pages 36–50. Springer, 2000.

[15] Bogdan Boteanu, Ionuţ Mironică, and Bogdan Ionescu. Pseudo-Relevance Feedback Diversification of Social Image Retrieval Results. *Multimedia Tools and Applications*, 76(9):11889–11916, 2017.

[16] Yue Cao, Mingsheng Long, Jianmin Wang, Qiang Yang, and Philip S Yu. Deep visual-semantic hashing for cross-modal retrieval. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1445–1454. ACM, 2016.

[17] David M Chen, Sam S Tsai, Vijay Chandrasekhar, Gabriel Takacs, Ramakrishna Vedantham, Radek Grzeszczuk, and Bernd Girod. Inverted Index Compression for Scalable Image Matching. In *Proceedings of the 2010 Data Compression Conference*, DCC'10, page 525, 2010.

[18] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A Large-Scale Audio-Visual Dataset. In *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP 2020, pages 721–725. IEEE, 2020.

[19] Yongjian Chen, Tao Guan, and Cheng Wang. Approximate Nearest Neighbor Search by Residual Vector Quantization. *Sensors*, 10(12):11259–11273, 2010.

[20] Yunqiang Chen, Xiang Sean Zhou, and Thomas S Huang. One-Class SVM for Learning in Image Retrieval. In *Proceedings of the 2001 International Conference on Image Processing*, pages 34–37. IEEE, 2001.

[21] Flavio Chierichetti, Alessandro Panconesi, Prabhakar Raghavan, Mauro Sozio, Alessandro Tiberi, and Eli Upfal. Finding near neighbors through cluster pruning. In *Proceedings of the 26th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS'07, pages 103–112, 2007.

[22] Kenneth Ward Church. Word2vec. *Natural Language Engineering*, 23(1):155–162, 2017.

[23] Pawel Cichosz. A Case Study in Text Mining of Discussion Forum Posts: Classification with Bag of Words and Global Vectors. *International Journal of Applied Mathematics and Computer Science*, 28(4), 2018.

[24] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995.

[25] Ryan R. Curtin and Andrew B. Gardner. Fast Approximate Furthest Neighbors with Data-Dependent Candidate Selection. In *Proceedings of the 9th International Conference on Similarity Search and Applications*, SISAP'16, pages 221–235. Springer, 2016.

[26] Duc-Tien Dang-Nguyen, Luca Piras, Giorgio Giacinto, Giulia Boato, and Francesco GB DE Natale. Multimodal Retrieval with Diversification and Relevance Feedback for Tourist Attraction Images. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 13(4):1–24, 2017.

[27] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. Locality-Sensitive Hashing Scheme based on P-Stable Distributions. In *Proceedings ACM Symposium on Computational Geometry*, pages 253–262. ACM, 2004.

[28] Thomas Deselaers, Tobias Gass, Philippe Dreuw, and Hermann Ney. Jointly Optimising Relevance and Diversity in Image Retrieval. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, CIVR'09, pages 1–8, 2009.

[29] Thomas Deselaers, Roberto Paredes, Enrique Vidal, and Hermann Ney. Learning Weighted Distances for Relevance Feedback in Image Retrieval. In *Proceedings of the 19th International Conference on Pattern Recognition*, ICPR'08, pages 1–4. IEEE, 2008.

[30] Shiv Ram Dubey. A Decade Survey of Content Based Image Retrieval Using Deep Learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(5):2687–2704, 2021.

[31] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Qian Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by Image and Video Content: The QBIC System. *Computer*, 28(9):23–32, 1995.

[32] Manuel J. Fonseca and Joaquim A. Jorge. Indexing High-Dimensional Data for Content-Based Retrieval in Large Databases. In *Proceedings of the 8th International Conference on Database Systems for Advanced Applications*, DASFAA'03, pages 267–274, 2003.

[33] Mingfei Gao, Zizhao Zhang, Guo Yu, Sercan Ö. Arık, Larry S. Davis, and Tomas Pfister. Consistency-Based Semi-supervised Active Learning: Towards Minimizing Labeling Cost. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Proceedings of the 16th European Conference on Computer Vision*, ECCV'20, pages 510–526. Springer, 2020.

[34] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized Product Quantization for Approximate Nearest Neighbor Search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR'13, pages 2946–2953, 2013.

[35] Philippe-Henri Gosselin and Matthieu Cord. Active Learning Techniques for User Interactive Systems: Application to Image Retrieval. In *International Workshop on Machine Learning techniques for Processing MultiMedia cCntent*, 2005.

[36] Gylfi Þór Guðmundsson, Laurent Amsaleg, Björn Þór Jónsson, and Michael J Franklin. Towards Engineering a Web-Scale Multimedia Service: A Case

Study Using Spark. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, MMSys'17, pages 1–12, 2017.

[37] Cathal Gurrin, Klaus Schoeffmann, Hideo Joho, Andreas Leibetseder, Liting Zhou, Aaron Duane, Dang Nguyen, Duc Tien, Michael Riegler, Luca Piras, et al. Comparing Approaches to Interactive Lifelog Search at the Lifelog Search Challenge (LSC2018). *ITE Transactions on Media Technology and Applications*, 7(2):46–59, 2019.

[38] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR'18, pages 6546–6555, 2018.

[39] Silvan Heller, Rahel Arnold, Ralph Gasser, Viktor Gsteiger, Mahnaz Parian-Scherb, Luca Rossetto, Loris Sauter, Florian Spiess, and Heiko Schuldt. Multi-Modal Interactive Video Retrieval with Temporal Queries. In *Proceedings of the 28th International Conference on Multimedia Modeling*, MMM'22, pages 493–498. Springer, 2022.

[40] Gisli R Hjaltason and Hanan Samet. Incremental Distance Join Algorithms for Spatial Databases. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, SIGMOD'98, pages 237–248. ACM, 1998.

[41] Danfeng Hong, Lianru Gao, Naoto Yokoya, Jing Yao, Jocelyn Chanussot, Qian Du, and Bing Zhang. More diverse means better: Multimodal deep learning meets remote-sensing imagery classification. *IEEE Transactions on Geoscience and Remote Sensing*, 59(5):4340–4354, 2020.

[42] Peisen S. Huang, Qingying Hu, Feng Jin, and Fu-Pen Chiang. Color-Encoded Digital Fringe Projection Technique for High-Speed 3-D Surface Contouring. *Optical Engineering*, 38(6):1065–1071, 1999.

[43] Thomas S Huang, Charlie K Dagli, Shyamsundar Rajaram, Edward Y Chang, Michael I Mandel, Graham E Poliner, and Daniel PW Ellis. Active Learning for Interactive Multimedia Retrieval. *Proceedings of the IEEE*, 96(4):648–667, 2008.

[44] Derry Jatnika, Moch Arif Bijaksana, and Arie Ardiyanti Suryani. Word2Vec Model Analysis for Semantic Similarities in English Words. *Procedia Computer Science*, 157:160–167, 2019.

[45] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product Quantization for Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2010.

[46] Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao. Latent Dirichlet allocation (LDA) and Topic Modeling: Models, Applications, a Survey. *Multimedia Tools and Applications*, 78(11):15169–15211, 2019.

[47] Lu Jiang, Teruko Mitamura, Shoou-I Yu, and Alexander G. Hauptmann. Zero-Example Event Search Using MultiModal Pseudo Relevance Feedback. In *Proceedings of the 4th ACM International Conference on Multimedia Retrieval*, ICMR'14, page 297–304. ACM, 2014.

[48] Björn Þór Jónsson, Marcel Worring, Jan Zahálka, Stevan Rudinac, and Laurent Amsaleg. Ten Research Questions for Scalable Multimedia Analytics. In *Proceedings of the 22nd International Conference on Multimedia Modeling*, MMM'16, pages 290–302. Springer, 2016.

[49] Björn Þór Jónsson, Omar Shahbaz Khan, Dennis C. Koelma, Stevan Rudinac, Marcel Worring, and Jan Zahálka. Exquisitor at the Video Browser Showdown 2020. In *Proceedings of the 26th International Conference on Multimedia Modeling*, MMM'20, page 796–802. Springer, 2020.

[50] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-Fusion: Audio-Visual Temporal Binding for Egocentric Action Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, ICCV'19, pages 5492–5501, 2019.

[51] Omar Shahbaz Khan, Björn Þór Jónsson, Mathias Larsen, Liam Poulsen, Dennis C. Koelma, Stevan Rudinac, Marcel Worring, and Jan Zahálka. Exquisitor at the Video Browser Showdown 2021: Relationships Between Semantic Classifiers. In *Proceedings of the 27th International Conference on Multimedia Modeling*, MMM'21, page 410–416. Springer, 2021.

[52] Omar Shahbaz Khan, Björn Þór Jónsson, Jan Zahálka, Stevan Rudinac, and Marcel Worring. Exquisitor at the Lifelog Search Challenge 2019. In *Proceedings of the 2nd Annual Workshop on Lifelog Search Challenge*, LSC'19, page 7–11. ACM, 2019.

[53] Omar Shahbaz Khan, Björn Þór Jónsson, Jan Zahálka, Stevan Rudinac, and Marcel Worring. Impact of Interaction Strategies on User Relevance Feed-

back. In *Proceedings of the International Conference on Multimedia Retrieval*, ICMR'21, page 590–598. ACM, 2021.

[54] Omar Shahbaz Khan, Björn Þór Jónsson, Stevan Rudinac, Jan Zahálka, Hanna Ragnarsdóttir, Þórhildur Þorleiksdóttir, Gylfi Þór Guðmundsson, Laurent Amsaleg, and Marcel Worring. Interactive Learning for Multimedia at Large. In *Proceedings of the 42nd European Conference on Information Retrieval*, ECIR'20, pages 495–510. Springer, 2020.

[55] Omar Shahbaz Khan, Mathias Dybkjær Larsen, Liam Alex Sonto Poulsen, Björn Þór Jónsson, Jan Zahálka, Stevan Rudinac, Dennis Koelma, and Marcel Worring. Exquisitor at the Lifelog Search Challenge 2020. In *Proceedings of the 3rd Annual Workshop on Lifelog Search Challenge*, LSC'20, page 19–22. ACM, 2020.

[56] Emil Knudsen, Thomas Holstein Qvortrup, Omar Shahbaz Khan, and Björn Þór Jónsson. XQC at the Lifelog Search Challenge 2021: Interactive Learning on a Mobile Device. In *Proceedings of the 4th Annual on Lifelog Search Challenge*, LSC'21, page 89–93. ACM, 2021.

[57] Sotiris B Kotsiantis, Ioannis Zaharakis, P Pintelas, et al. Supervised Machine Learning: A Review of Classification Techniques. *Emerging Artificial Intelligence Applications in Computer Engineering*, 160(1):3–24, 2007.

[58] Miroslav Kratochvil, František Mejzlík, Patrik Veselý, Tomáš Soućek, and Jakub Lokoć. *SOMHunter: Lightweight Video Search System with SOM-Guided Relevance Feedback*, page 4481–4484. MM'20. ACM, 2020.

[59] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, NIPS'12. Curran Associates Inc., 2012.

[60] Eyal Kushilevitz, Rafail Ostrovsky, and Yuval Rabani. Efficient Search for Approximate Nearest Neighbor in High Dimensional Spaces. *SIAM Journal on Computing*, 30(2):457–474, 2000.

[61] Herwig Lejsek, Friðrik Heiðar Ásmundsson, Björn Þór Jónsson, Laurent Amsaleg, et al. NV-Tree: An Efficient Disk-based Index for Approximate Search in Very Large High-Dimensional Collections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):869–883, 2009.

[62] Herwig Lejsek, Ársæll Þ Jóhannsson, Friðrik H Ásmundsson, Björn Þ Jónsson, Kristleifur Daðason, and Laurent Amsaleg. Videntifier™ Forensic: A

New Law Enforcement Service for Automatic Identification of Illegal Video Material. In *Proceedings of the 1st ACM Workshop on Multimedia in Forensics*, pages 19–24, 2009.

[63] Herwig Lejsek, Björn Þór Jónsson, and Laurent Amsaleg. NV-Tree: Nearest Neighbors at the Billion Scale. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, ICMR'11. ACM, 2011.

[64] Dan Li and Evangelos Kanoulas. When to Stop Reviewing in Technology-Assisted Reviews: Sampling from an Adaptive Distribution to Estimate Residual Relevant Documents. *ACM Transactions on Information Systems*, 38(4), 2020.

[65] Lin Tzy Li, Daniel Carlos Guimarães Pedronette, Jurandy Almeida, Otávio AB Penatti, Rodrigo Tripodi Calumby, and Ricardo da Silva Torres. A Rank Aggregation Framework for Video Multimodal Geocoding. *Multimedia Tools and Applications*, 73(3):1323–1359, 2014.

[66] Lizi Liao, Xiangnan He, Bo Zhao, Chong-Wah Ngo, and Tat-Seng Chua. Interpretable Multimodal Retrieval for Fashion Products. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM'18, pages 1571–1579, 2018.

[67] Qing Liu, Jing Wang, Dehai Zhang, Yun Yang, and NaiYao Wang. Text Features Extraction based on TF-IDF Associating Semantic. In *Proceedings of the IEEE 4th International Conference on Computer and Communications*, ICCC'18, pages 2338–2343, 2018.

[68] Zimo Liu, Jingya Wang, Shaogang Gong, Huchuan Lu, and Dacheng Tao. Deep Reinforcement Active Learning for Human-in-the-Loop Person Re-Identification. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, ICCV'19, 2019.

[69] David G. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the 7th IEEE International Conference on Computer Vision*, volume 2 of *ICCV'99*, pages 1150–1157, 1999.

[70] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[71] Qin Lv, William Josephson, Zhe Wang, Moses Charikar, and Kai Li. Multi-probe lsh: Efficient indexing for high-dimensional similarity search. In *Proceedings of the 33rd International Conference on Very Large Data Bases*, VLDB'07, pages 950–961. ACM, 2007.

[72] Qin Lv, William Josephson, Zhe Wang, Moses Charikar, and Kai Li. Intelligent Probing for Locality Sensitive Hashing: Multi-Probe LSH and Beyond. *Proceedings of the VLDB Endowment*, 2017.

[73] Bangalore S Manjunath and Wei-Ying Ma. Texture Features for Browsing and Retrieval of Image Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, 1996.

[74] Graham McDonald, Craig Macdonald, and Iadh Ounis. Active Learning Stopping Strategies for Technology-Assisted Sensitivity Review. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'20, page 2053–2056. ACM, 2020.

[75] Robert B Miller. Response Time in Man-Computer Conversational Transactions. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 267–277, 1968.

[76] Diana Moise, Denis Shestakov, Gylfi Gudmundsson, and Laurent Amsaleg. Indexing and Searching 100M Images with Map-Reduce. In *Proceedings of the 3rd ACM International Conference on Multimedia Retrieval*, ICMR'13, pages 17–24. ACM, 2013.

[77] Ngoc Duy Nguyen, Thanh Nguyen, and Saeid Nahavandi. System Design Perspective for Human-Level Agents Using Deep Reinforcement Learning: A Survey. *IEEE Access*, 5:27091–27102, 2017.

[78] Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H Adelson, and William T Freeman. Visually indicated sounds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR'16, pages 2405–2413, 2016.

[79] Rasmus Pagh, Francesco Silvestri, Johan Sivertsen, and Matthew Skala. Approximate Furthest Neighbor with Application to Annulus Query. *Information Systems*, 64:152–162, 2017.

[80] Geert Pingen, Maaike de Boer, and Robin Aly. Rocchio-Based Relevance Feedback in Video Event Retrieval. In *Proceedings of the 23rd International Conference on Multimedia Modeling*, MMM'17, pages 318–330. Springer, 2017.

[81] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack

Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, PMLR'21, pages 8748–8763. PMLR, 2021.

[82] H. Ragnarsdóttir, Þ. Þorleiksdóttir, O. S. Khan, B. Þ. Jónsson, G. Þ. Guðmundsson, J. Zahálka, S. Rudinac, L. Amsaleg, and M. Worring. Exquisitor: Breaking the interaction barrier for exploration of 100 million images. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM'19, pages 687–698. ACM, 2019.

[83] Juan Ramos. Using TF-IDF to Determine Word Relevance in Document Queries. In *Proceedings of the 1st Instructional Conference on Machine Learning*, volume 242, pages 29–48, 2003.

[84] William Ribarsky and Brian Fisher. The Human-Computer System: Towards an Operational Model for Problem Solving. In *Proceedings of the 49th Hawaii International Conference on System Sciences*, HICCS'16, pages 1446–1455, 2016.

[85] Ork De Rooij and Marcel Worring. Efficient Targeted Search Using a Focus and Context Video Browser. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 8(4), 2012.

[86] Yong Rui, Thomas S Huang, and Sharad Mehrotra. Content-Based Image Retrieval with Relevance Feedback in MARS. In *Proceedings of International Conference on Image Processing*, ICIP'97, pages 815–818. IEEE, 1997.

[87] Yong Rui, Thomas S Huang, Michael Ortega, and Sharad Mehrotra. Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):644–655, 1998.

[88] Amaia Salvador, Nicholas Hynes, Yusuf Aytar, Javier Marin, Ferda Ofli, Ingmar Weber, and Antonio Torralba. Learning cross-modal embeddings for cooking recipes and food images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR'17, pages 3020–3028, 2017.

[89] Hanan Samet. K-Nearest Neighbor Finding Using MaxNearestDist. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):243–252, 2008.

[90] Kim I. Schild, Alexandra M. Bagi, Magnus Holm Mamsen, Omar Shahbaz Khan, Jan Zahálka, and Björn Þór Jónsson. XQM: Search-Oriented vs. Classifier-Oriented Relevance Feedback on Mobile Phones. In *Proceedings of the 28th International Conference on Multimedia Modeling*, volume 13142 of *MMM'22*, pages 458–464. Springer, 2022.

[91] Klaus Schoeffmann. A User-Centric Media Retrieval Competition: The Video Browser Showdown 2012-2014. *IEEE MultiMedia*, 21(4):8–13, 2014.

[92] Lokesh Setia, Julia Ick, Hans Burkhardt, and AI Features. SVM-based Relevance Feedback in Image Retrieval using Invariant Feature Histograms. In *Proceedings of the IAPR Conference on Machine Vision Applications*, MVA'05, pages 542–545. Citeseer, 2005.

[93] Burr Settles. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[94] Jie Shao, Xin Wen, Bingchen Zhao, and Xiangyang Xue. Temporal Context Aggregation for Video Retrieval with Contrastive Learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, WACV'21, pages 3268–3278, 2021.

[95] Ajay Shrestha and Ausif Mahmood. Review of Deep Learning Algorithms and Architectures. *IEEE Access*, 7:53040–53065, 2019.

[96] Cees GM Snoek, Marcel Worring, and Arnold WM Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th ACM International Conference on Multimedia*, MM'05, pages 399–402, 2005.

[97] Michael J. Swain and Dana H. Ballard. Color Indexing. *International Journal of computer Vision*, 7(1):11–32, 1991.

[98] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, Inception-Resnet and the Impact of Residual Connections on Learning. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, AAAI'17, 2017.

[99] Simon Tong and Edward Chang. Support Vector Machine Active Learning for Image Retrieval. In *Proceedings of the 9th ACM International Conference on Multimedia*, MM'01, pages 107–118, 2001.

[100] Roberto Tronci, Gabriele Murgia, Maurizio Pili, Luca Piras, and Giorgio Giacinto. *ImageHunter: A Novel Tool for Relevance Feedback in Content Based Image Retrieval*, pages 53–70. Springer, 2013.

[101] Tinne Tuytelaars and Krystian Mikolajczyk. A Survey on Local Invariant Features. *Foundations and Trends in Computer Graphics and Vision*, 3(3):176–280, 2008.

[102] Sudheendra Vijayanarasimhan, Prateek Jain, and Kristen Grauman. Hashing Hyperplane Queries to Near Points with Applications to Large-Scale Active Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):276–288, 2014.

[103] Mandikal Vikram, Aditya Anantharaman, and Suhas BS. An Approach for Multimodal Medical Image Retrieval Using Latent Dirichlet Allocation. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, pages 44–51, 2019.

[104] Huayi Wang, Jingfan Meng, Long Gong, Jun Xu, and Mitsunori Ogihara. MP-RW-LSH: an efficient multi-probe LSH solution to ANNS-L 1. *In Proceedings of the VLDB Endowment*, 14(13):3267–3280, 2021.

[105] Jenq-Haur Wang, Ting-Wei Liu, Xiong Luo, and Long Wang. An LSTM Approach to Short Text Sentiment Classification with Word Embeddings. In *Proceedings of the 30th Conference on Computational Linguistics and Speech Processing*, ROCLING'18, pages 214–223, 2018.

[106] Xuanhui Wang, Hui Fang, and ChengXiang Zhai. A Study of Methods for Negative Relevance Feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'08, page 219–226. ACM, 2008.

[107] Hao Yan, Shuai Ding, and Torsten Suel. Inverted Index Compression and Query Processing with Optimized Document Ordering. In *Proceedings of the 18th International Conference on World Wide Web*, WWW'09, pages 401–410, 2009.

[108] Rong Yan, Alexander Hauptmann, and Rong Jin. Multimedia Search with Pseudo-Relevance Feedback. In *Image and Video Retrieval*, pages 238–247. Springer, 2003.

[109] Rong Yan, Alexander G. Hauptmann, and Rong Jin. Negative Pseudo-Relevance Feedback in Content-Based Video Retrieval. In *Proceedings of the 11th ACM International Conference on Multimedia*, MM'03, page 343–346. ACM, 2003.

[110] Donggeun Yoo and In So Kweon. Learning Loss for Active Learning. In *Proceedings of the EEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR'19, 2019.

[111] Jan Zahálka, Stevan Rudinac, Björn Þór Jónsson, Dennis C Koelma, and Marcel Worring. Blackthorn: Large-Scale Interactive Multimodal Learning. *IEEE Transactions on Multimedia*, 20(3):687–698, 2018.

[112] Jan Zahálka, Stevan Rudinac, and Marcel Worring. Analytic quality: Evaluation of performance and insight in multimedia collection analysis. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM'15, page 231–240, New York, NY, USA, 2015. ACM.

[113] Jan Zahálka and Marcel Worring. Towards Interactive, Intelligent, and Integrated Multimedia analytics. In *Proceedings of 2014 IEEE Conference on Visual Analytics Science and Technology*, pages 3–12. IEEE, 2014.

[114] Jan Zahálka, Marcel Worring, and J. J. Van Wijk. II-20: Intelligent and Pragmatic Analytic Categorization of Image Collections. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):422–431, 2021.

[115] Wenyi Zhao, Rama Chellappa, P. Jonathan Phillips, and Azriel Rosenfeld. Face Recognition: A Literature Survey. *ACM Computing Survey*, 35(4):399–458, 2003.

[116] Liang Zheng, Shengjin Wang, Lu Tian, Fei He, Ziqiong Liu, and Qi Tian. Query-Adaptive Late Fusion for Image Search and Person Re-Identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR'15, pages 1741–1750, 2015.

[117] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 Million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[118] Xiang Zhou and Thomas Huang. Relevance Feedback in Image Retrieval: A Comprehensive Review. *Multimedia Systems*, 8:536–544, 2003.