Blackthorn Pruning: High-Performance Interactive Multimodal Learning with Approximate High-Dimensional Indexing

by

Hanna Ragnarsdóttir
Þórhildur Þorleiksdóttir

Thesis of 12 ECTS credits submitted to the School of Computer Science
at Reykjavík University in partial fulfillment
of the requirements for the degree of
Bachelor of Science (B.Sc.) in Computer Science

May 2018

Examining Committee:

Dr. Björn Þór Jónsson, Supervisor
Associate Professor, Reykjavík University, Iceland
Associate Professor, IT University of Copenhagen, Denmark

Dr. Gylfi Þór Guðmundsson, Co-advisor
Adjunct Professor, Reykjavík University, Iceland

Dr. Yngvi Björnsson, Examiner
Professor, Reykjavík University, Iceland

Blackthorn Pruning: High-Performance Interactive Multimodal Learning with Approximate High-Dimensional Indexing

Hanna Ragnarsdóttir
Þórhildur Þorleiksdóttir

May 2018

Abstract

As the size of multimedia collections grows, so does the need for efficient and scalable search and exploration methods. In this thesis we present Blackthorn Pruning, a scalable, interactive multimodal learning approach that facilitates interactive analysis of vast multimedia collections. Blackthorn Pruning is created by combining Blackthorn, a state-of-the-art interactive multimodal learning approach, and eCP, a scalable, approximate high-dimensional indexing method. By pruning Blackthorn's search space, eCP reduces the number of data items scored by Blackthorn in each interaction round, leading to reduced time per interaction round, while maintaining the relevance of the items suggested. Experiments on the YFCC100M dataset, which consists of nearly 100 million images and metadata, show that compared to original Blackthorn, Blackthorn Pruning takes 14 times less time using 16 times less computational power. Experiments on a simulated collection of 1 billion images also further suggest the scalability potential of Blackthorn Pruning. Our proposed approach thus opens up interesting avenues for analytics on truly Web-scale collections and also unlocks the potential for such analytics to be performed on modest hardware configurations commonly found in consumer PCs and mobile devices.

"Space is big. You just won't believe how vastly, hugely, mind- bogglingly big it is. I mean, you may think it's a long way down the road to the chemist's, but that's just peanuts to space." - The Hitchhiker's Guide to the Galaxy

Acknowledgements

We want to thank Reykjavík University for providing us facilities while working on this project. Thanks to Dennis Koelma for giving us access to DAS-5 server at the University of Amsterdam. Jan Zahálka, Stevan Rudinac, Laurent Amsaleg, and Marcel Worring we want to thank for reviewing our work, their contribution, guidance, assistance, and motivation. Thank you Gylfi Þór Guðmundsson for always being ready to help no matter what time it is. Lastly, we want to thank Björn Þór Jónsson for believing in us and pushing us to the limits.

Sorry, mom and dad for never being home.

Preface

Part of this thesis is built on a research paper submitted to the ACM Multimedia Conference April 8th 2018, jointly written by the students, supervisors and authors of Blackthorn and eCP.

The students' contribution to the research paper submitted was conducting the research, coding the program, participating in all meetings, doing all experiments, analysing results and writing the text in Chapter 4 (Blackthorn and eCP) which describes Blackthorn Pruning. The remainder of the text was written by co-authors.

In this thesis, chapters 1, 2 and 4 are based on the research paper. Chapters 3, 5 and 6 in this report are original and independent work of the students.

Contents

List of Figures

List of Tables

1 Introduction

Multimedia collections are becoming a central information resource for a growing number of domains, including satellite data, digital forensics, healthcare, social media, cultural heritage and various industrial applications. Analysing large-scale multimedia collections, which can range from millions to billions of media items, to retrieve information is becoming more vital with ever-growing data. The constant growth of data collections requires fast and scalable methods to gain insightful knowledge from the data.

Recently, analysing large-scale multimedia collections interactively has been proposed as an important aspect of multimedia analytics [1]. Analysing multimedia collections interactively allows the user to categorize the data, resulting in a custom analyst-driven data categorization, unlike general state-of-the-art search engines. To enable users to explore large-scale multimedia collections, the interaction between the user and the system must execute efficiently, preferably within sub-second response time, since users are becoming ever more demanding.

Blackthorn, an interactive multimodal learning algorithm, was recently proposed as a methodology to extract knowledge from large-scale multimedia collections [2]. Experiments with Blackthorn have demonstrated that the algorithm can learn user-preferences for 100 million images in just over 1 second while providing better results than state-of-the-art relevance feedback algorithms that take minutes for the same task. As far as we are aware, Blackthorn is the most scalable image analysis and retrieval system available today. Even this performance may not be sufficient for the large-scale interactive applications of the future, like we will discuss in Section 2.1.

Approximate high-dimensional indexing, adapted to the Blackthorn compression format, could be used to speed up performance, provided that the quality of the approximate results is sufficient. One such indexing method is the extended Cluster Pruning algorithm, eCP [3], which has been shown to give results of good quality with limited processing and has also been shown to scale well to massive collections.

In this thesis, we considerably improve the efficiency of large-scale interactive multimodal learning by applying approximate high-dimensional indexing to Blackthorn. In the process, we make the following scientific contributions:

1. We discuss and analyse the weak point of interactive multimodal learning on large-scale data, i.e., the amount of data unnecessarily considered in the scoring process (Chapter 2).

2. We describe the data access requirements of interactive multimodal learning, and analyse the applicability of high-dimensional indexing method, eCP (Chapter 2).

3. We propose a new method that significantly improves the performance of interactive multimodal learning using eCP (Chapter 3).

4. We evaluate the new method on large-scale dataset, demonstrating state-of-the-art accuracy of results using dramatically reduced computational resources (Chapter 4).

5. We discuss the limitations of our method and describe future work (Chapter 5).

Overall, our experiments with the YFCC100M dataset show that state-of-the-art results are retrieved in less than 0.08 seconds using a single CPU-core, or about 14x less time using 16x less computational power, opening up avenues for full system integration, mobile applications, and more rapid interaction. Finally, we demonstrate the scalability potential of our system using a simulated collection of one billion images.

2 Background

In this chapter, we describe the two techniques Blackthorn Pruning is based on, Blackthorn and eCP, as well as discussing related work in the field of interactive multimodal learning and high-dimensional indexing.

2.1 Interactive Multimodal Learning

Interactive multimodal learning, which has existed since the 2000s [4], has recently been proposed as an important component of multimedia analytics [1]. The goal of interactive multimodal learning is to enable interactive analytics over multimedia collections, without forcing the analyst to explicitly formulate a query. Although users may have difficulties in formulating a precise query for a specific task, they generally see quickly whether a returned result is relevant to the information need or not. Interactive multimodal learning thus allows users to categorize the media items, resulting in a custom analyst-driven data categorization. Interactive multimodal learning has been used in a number of systems, e.g., City Melange [5] and Informedia Digital Video Library [6]. In addition, interactive multimodal learning has been in the spotlight of benchmarks and challenges such as VideOlympics [7] and the Video Browser Showdown [8]. In recent years, the size of multimedia collections has been increasing dramatically, leading to a sharp decrease in the performance of state-of-the-art interactive multimodal learning systems. At the same time, the call for user-focused, personalized approaches has been intensifying and Blackthorn [2] has answered this call by revising interactive multimodal learning to work on very large scale collections. Experiments with Blackthorn demonstrated that the algorithm can learn user preferences over 100 million images in just over one second per interaction round, using a high-end workstation with 16 computing cores, while providing better results than state-of-the-art relevance feedback algorithms that took minutes for the same task [2]. Blackthorn was also shown to significantly outperform Product Quantization [9], which is a pure k-NN-based approach. Blackthorn has thus been demonstrated as the dominant approach in the interactive, semi-supervised setting.

2.2 Blackthorn

As already mentioned, Blackthorn is an efficient interactive multimodal learning approach facilitating analysis of multimedia collections of 100 million items. This performance is achieved by efficient data compression and optimizations to the state-of-the-art interactive learning process. Blackthorn's compressed data representation, called Ratio-64 representation, preserves only the most significant features for each item, discarding the others. It requires only $2l + 1$ 64-bit integers per item and modality, where l is a parameter to control the extent of the compression. For $l = 1$, it takes up to 183.5x less memory than the uncompressed features, whilst preserving or even improving precision in interactive multimodal learning. Figure 2.1 represents the three 64-bit integers that are used to represent the top features of one item in the

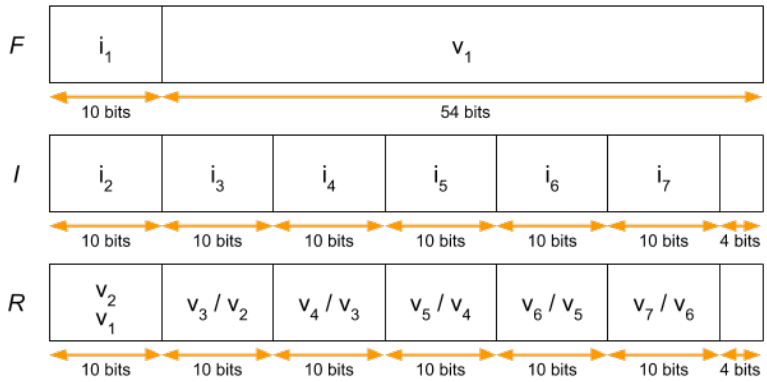

Figure 2.1: The ratio representation of one item. The strongest feature of an item is represented with F, I is feature identifiers, and R encodes ratios between subsequent features.

compressed representation, when $l = 1$. The first integer (F) is the strongest feature of the corresponding item. The first 10 bits of F encode the feature identifier, and last 54 bits encode the feature's value. The second integer (I) encodes the feature identifiers of the remaining top items while the third integer (R) encodes the ratio between the current item and previous item. For I, 10 bits represent each identifiers and for R, 10 bits represent each ratio of subsequent values of each feature, leaving 4 bits unused. Thus, when $l = 1$ the ratio representation compresses 7 top features of each item into 64-bit integers F, I, and R. The integers are used for calculation in a compressed space by decompressing the ratios on the fly using the strongest feature value. The identifiers of the features for one item range from 0 to 999, since each item is assigned at most 1000 features, which depends on its modality. The optimized interactive learning model then scores the compressed data directly, greatly reducing the computational requirements [2].

The pipeline of the interactive multimodal learning process for Blackthorn can be seen in Figure 2.2. After feature extraction of the data, the semantic representation is compressed and the compressed representation is used for the rest of the process. In the experiments for Blackthorn we seperate feature extraction methods for two modalitites, visual and textual modalities. For visual modality 1000 ILSVRC visual concepts [10] are extracted from the images using a GoogLeNet convolutional neural network [11]. For textual modality, 100 LDA concepts are extracted using gensim toolkit [12]. In this setting, the user interacts with the machine in interaction rounds. Let M denote the data collection, with $N = |M|$. In each interaction round i, the user is presented with $\mathcal{S}_i \subset M$, a set of items suggested by the system. Interactive multimodal learning operates in a semi-supervised setting, i.e., $\forall i,\ |\mathcal{S}_i| \ll N$. The user then passes judgement on the items in $\mathcal{S}_i$, marking the relevant and non-relevant ones. The system then takes these judgements, trains an interactive classifier, scores the items in M, and produces $\mathcal{S}_{i+1}$, the suggested items for the next interaction round. When scoring the items, Blackthorn employs heavy parallelization. For maximum efficiency Blackthorn requires 16 CPU cores. In that case each worker scores $\frac{N}{16}$ items and stores all of its scores in a list, which can be used to change the scores of items seen in previous rounds to a high negative value. Thus, items already seen are not considered for subsequent rounds. The training and scoring process is multimodal, operating on visual and text features. Blackthorn uses late modality fusion to combine

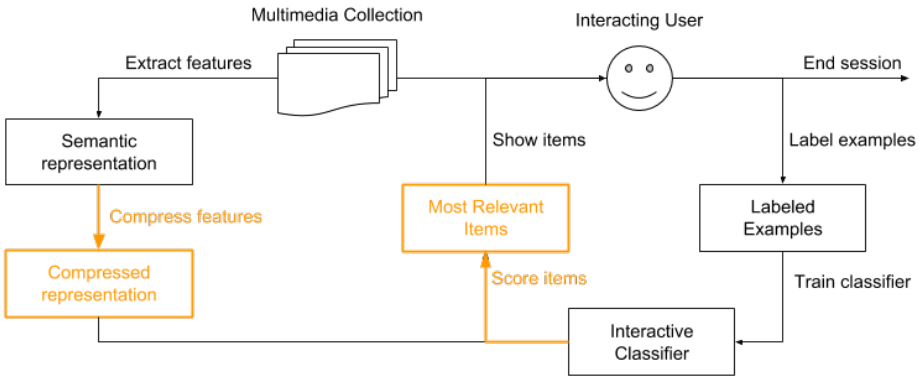

Figure 2.2: Blackthorn's pipeline of interactive learning. The orange components are interactive learning methods innovated by Blackthorn [2].

the modalities, which has been established to perform better than early fusion [13]. The system trains an interactive classifier for each modality, scores and ranks the items in each modality separately, and then fuses the rankings into the final ranking based on which $\mathcal{S}_{i+1}$ is constructed. Each worker performs modality fusion and returns a single list of its top items, which is then combined into one list by the main process.

As far as we are aware, Blackthorn is the most efficient image analytics system available today. However, even this performance may not be sufficient for the analytics applications of the future for the following reasons:

1. The interactive multimodal learning process may be embedded in a larger system, which requires computing resources for other processes. While Blackthorn can run with fewer computing resources, the response time would increase correspondingly, thus severely limiting interactivity.

2. Running interactive multimodal learning on mobile platforms would significantly improve the accessibility of such applications. However, Blackthorn's extensive computing requirements preclude its use on mobile platforms.

3. Even one second per interaction round may not be sufficiently fast for many analysts, as users become ever more demanding of the interaction with computing systems. Reducing the response time significantly is thus important, especially in the context above of reduced access to computational power

4. As multimedia collections are ever growing, interactive multimodal learning systems may in some cases be called upon to handle collections that are significantly larger than the YFCC100M dataset.

In this thesis, we address all these factors.

2.3 High-Dimensional Indexing

As already mentioned in Section 2.2, Blackthorn suggests $\mathcal{S}_i \subset M$ items (which we from now on refer to as k items) to the analyst per interaction round, and then uses the feedback from the analyst to modify its model of the classification boundary. Each time it suggests the items most likely to be relevant, i.e., those furthest from the

classification boundary. The interaction with the image feature collection thus amounts to a k-farthest neighbour query. Due to the curse of dimensionality, scalable high-dimensional indexing methods must rely on approximate similarity searches, typically trading-off small reductions in quality (or even just quality guarantees) for dramatic response time improvements. While k-farthest neighbour queries have been studied in the literature, those works have only considered the distance from point queries, rather than the distance from classification boundaries. We are not aware of any work in the high-dimensional literature specifically targeting approximate k-FN where the query is a classification boundary.

Analysis of the processing pipeline of Blackthorn yields the following requirements for a successful high-dimensional indexing approach.

1. *Stable Response Time:* The highly interactive nature of the process demands a short and predictable response time, to avoid distracting the analysts. The approximate nature of the queries and features, on the other hand, limits the impact of any quality guarantees in the high-dimensional space. Response time guarantees are thus far more valuable than result quality guarantees [14].

2. *Preservation of Feature Space Similarity Structure:* The interactive classifiers should capture the user intent as it evolves during the interactive session. The interactive classifiers compute relevance of individual items based on the similarity structure of the feature space. The space partitioning of the high-dimensional indexing algorithm must follow and preserve this similarity structure.

3. *k Farthest Neighbors:* Blackthorn requests the items farthest from the classification boundary. Furthermore, as the results are intended for display on screen, the index must return exactly k-farthest neighbours (k-FN). Finally, since the interactive classifier is an approximation of the analyst's intent, approximate answers are also acceptable.

In next section we will show that that extended Cluster Pruning (eCP) [3], an approximate cluster-based high-dimensional algorithm, is a very promising candidate that fulfills these requirements.

2.4 Extended Cluster Pruning

eCP clusters a feature collection with a vectorial quantizer, using a deep hierarchical index structure. Clusters are formed by randomly picking data points from the collection, called representatives, and then assigning all the remaining points to these representatives based on proximity (essentially the first step of the k-means algorithm). Then, at query time, the representative that is the closest to the query point is determined by traversing the index, and the distances to all points in that cluster are computed. To compensate for the naïve clustering strategy, search expansion (more than one cluster is scanned at query time) and soft-assignments (data point get assigned to multiple clusters) can be used, see [15] for further details on both methods. eCP was proposed for very large-scale collections, where disk-based processing is necessary. In that context, both small and large clusters hurt performance: small clusters require expensive disk reads for little gains in quality, while large clusters require extensive computing to score [16]. Unlike most clustering algorithms, eCP therefore aims to keep the size

of clusters as even as possible, yielding excellent approximate results with small and predictable response times. When processing extremely large collections the number of clusters created must grow accordingly. In turn, however, clusters are organized in a hierarchy such that the cost of identifying the most relevant clusters remains efficient. Both eCP, and its distributed variant DeCP [17], [18], have been extensively evaluated on high dimensional datasets at a very large scale (millions of clusters with billions of feature vectors).

2.5 Summary

In this chapter we have highlighted four reasons why Blackthorn's performance may not satisfy the analytics applications of the future. To address those computational and scalability issues we propose Blackthorn Pruning, discussed in next chapter, which integrates eCP's approximate search method and deep index hierarchy into Blackthorn's process of scoring the compressed representation data.

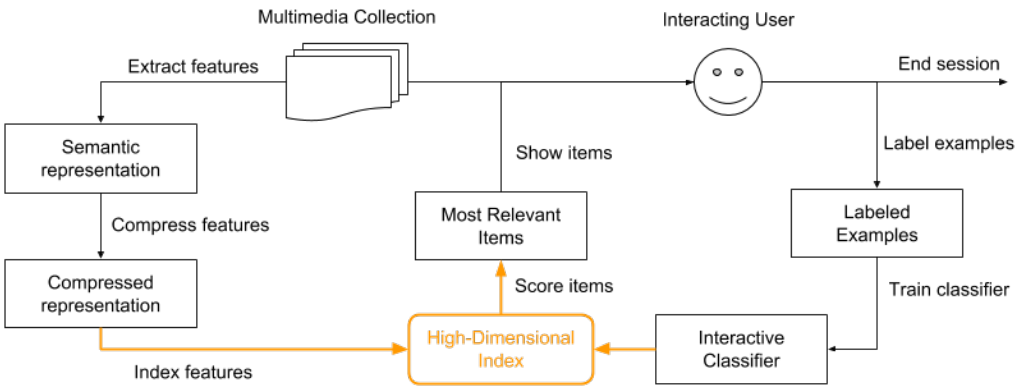

Figure 3.1: Interactive Multimodal Learning pipeline integrated with High-Dimensional Indexing. The black components are based on the original Blackthorn design and the orange components represent the integration of the indexing method of eCP.

3 Blackthorn Pruning

In this chapter, we highlight the main modifications required for the integration of eCP and Blackthorn, focusing on the indexing phase of the Ratio-64 compressed data and the scoring of the clustered compressed data.

3.1 System Design Overview

Figure 3.1 shows the design of Blackthorn Pruning. As previously mentioned, Blackthorn Pruning is based on the original system design of Blackthorn, described in Figure 2.2. The differences between Blackthorn Pruning and the original Blackthorn can be divided into two main parts:

1. Clustering of the compressed Ratio-64 representation, using the indexing method of eCP, described in Section 3.2.

2. Scoring of the indexed data representation in the interactive learning cycle, explained in Section 3.3.

Blackthorn is written in C and eCP is written in C++. The integrated system, Blackthorn Pruning, is written in C++ and compiled with g++. Python scripts are used for data manipulation and to set up experiments. All experiments were done on the DAS-5 (The Distributed ASCI Supercomputer 5) server. DAS-5 includes roughly 200 dual-eight-core compute nodes spread out over six clusters, located at five sites. The site we used was at the University of Amsterdam (UvA). The UvA site server has a 64 GB RAM and 4TB local SSD. The operating system of DAS-5 server is CentOS Linux. The SLURM resource management system was used to reserve requested number of nodes on the DAS-5 when doing experiments [19]. Since all development was performed

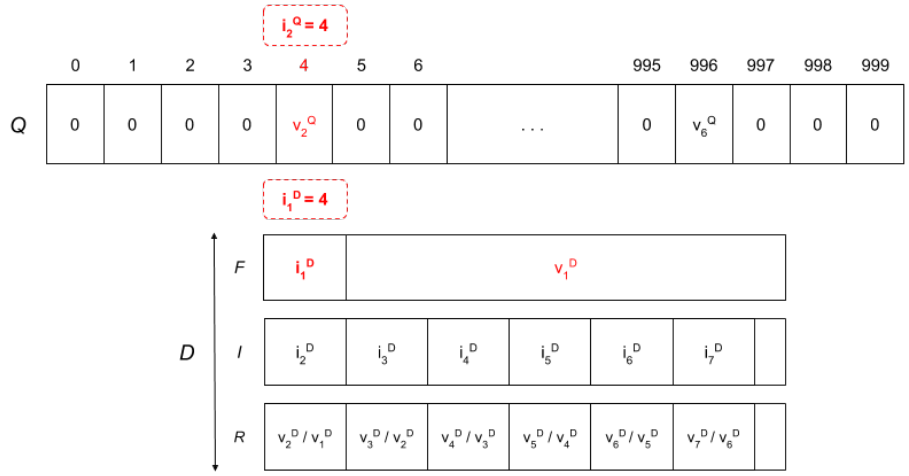

Figure 3.2: Distance is calculated between a 1000 dimensional decompressed query descriptor and a compressed data item. E.g. if the top feature identifier of D is 4, the $4th$ element of Q is accessed to get the corresponding value, v_2^Q, which is then used in the Euclidean distance formula along with v_1^D.

on the server, vim was used for writing and editing code. All code is maintained using git version control.

3.2 Indexing the Ratio-64 Representation

The index for the compressed representation is constructed using the indexing methodology of eCP. Each phase is adapted to the compressed representation of a descriptor, which is three 64-bit integers as described in Section 2.2. The Euclidean distance function is adapted to the compressed representation and used as the discriminative distance function for eCP. The distance is calculated between representatives when constructing the index and between representatives and other items when assigning items to clusters. Representatives are referred to as query descriptors when computing distance. The query descriptors need to be decompressed for distance calculations.

Figure 3.2 shows a decompressed query descriptor, which is represented as a 1000 element array Q, initialized with values equal to 0. The array is constructed using the identifiers from the compressed query descriptor and corresponding indices of the array to place the values to its original place. Those indices that do not have a matching identifier from the query descriptor are assigned a value of 0, since most of the original feature values are negligible.

Figure 3.2 also shows how distance is computed between the query descriptor and a compressed data item. First, the identifiers of D are traversed and its decompressed values are calculated. Then, each identifier of D is used as an index into Q to find the value for the same feature identifier in Q (which is 0 if it was not one of the top features for Q). The values are used as an input to the Euclidean distance formula ($n = 7$).

$$Distance = \sum_{k=0}^{n-1} (Q[i_k^D] - v_k^D)^2 \tag{3.1}$$

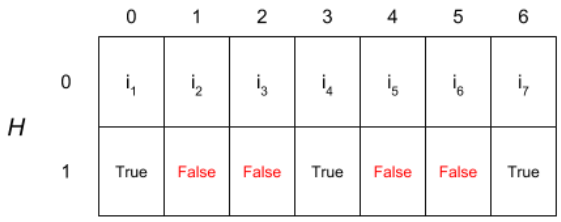

Figure 3.3: Two-dimensional array H, which holds information regarding whether a query descriptor and a data item have a matching feature identifier.

When computing the distance between elements we maintain a two-dimensional array H, shown in Figure 3.3, which holds information regarding whether the query descriptor and the data item have matching feature identifiers. The first row of the array holds the identifiers of the query descriptor Q, while the elements of the second row indicate whether the corresponding identifier is present in D (initialized with *False*). If an identifier of Q is found in D, the value for corresponding identifier in H is changed to *True*. Thus feature identifiers of Q that were present in D are labeled with *True*, while other identifiers are labeled with *False*. When we have finished traversing the identifiers of D, we traverse H to add values of feature identifiers in Q that were not present in D. Thus, if $H[1][i] = False$ for each $i = 0 \dots (n-1)$, where $n = 7$, we calculate the results according to Function 3.2.

$$Distance = Distance + \sum_{k=0}^{n-1} \begin{cases} Q[H[0][i]]^2, & \text{if } H[1][i] = False \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

The use of two modalities further demands two separate indices. Thus, the indexing is executed once for the visual modality and once for the textual modality.

When building indices, the average cluster size was chosen to be small since searching more small clusters yields better results than searching fewer large clusters [16]. Smaller clusters are also faster to process.

3.3 Scoring Clustered Data

To reduce the overhead of scoring the whole dataset, we replace Blackthorn's default policy of scoring everything in each round with an approximation policy that only evaluates a relevant subset of data, as identified by the approximate eCP search algorithm. If the indexed high-dimensional collection for each modality fits in memory, it is loaded at the start of a session, otherwise the clusters are loaded dynamically from disk. The scoring of the clustered data then proceeds in the following three steps:

1. Find the top b clusters
2. Find the top k results
3. Merging top k results from both modalities

Below we detail each of these steps.

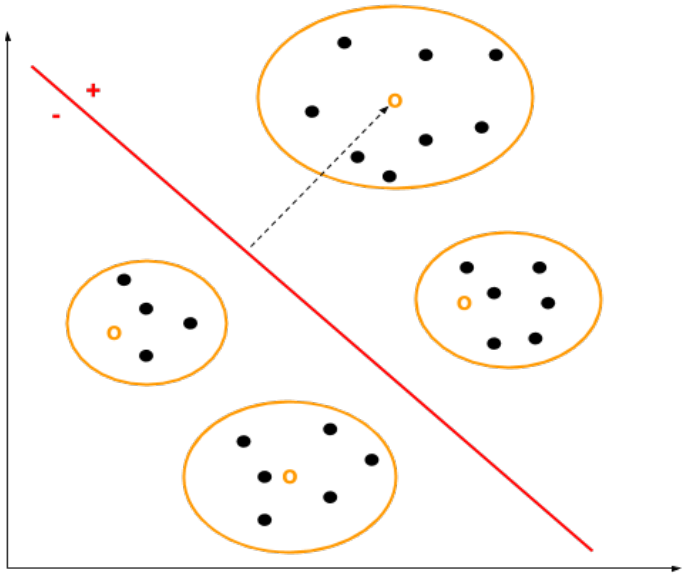

Figure 3.4: Simplified idea of finding top b clusters. The red line represents a classification boundary. To find top b clusters, distance is calculated between the boundary and all representatives, and b clusters with the largest distance in positive direction are selected. The dotted line represents the distance to the farthest (best) cluster.

3.3.1 Find the Top b Clusters

The number of all clusters, according to the given indexed representation, is denoted with C. The number of top clusters considered can be adjusted by a search expansion parameter b, which affects the size of the subset that will be scored. This parameter can be used to balance between search quality and performance at runtime. For higher values of b, more clusters are considered and thus a larger subset of the data will be scored. When $b = C$ all of the data will be scored.

In each round, after training an interactive classifier, the representatives of all clusters are given a score of relevance, by computing the distance from the representative to the classification boundary. This process is described in Figure 3.4. The classifier used in our experiments is a linear SVM and the distance computations are done directly in the compressed space, in a similar way as in Blackthorn: The dot-product between each representative and the SVM separating plane is calculated and the b clusters farthest from the plane (in the positive direction) are selected as the most relevant clusters.

3.3.2 Find the Top k Results

Items within the top b clusters are scored to suggest top k results, which significantly reduces the amount of data considered. When using multiple workers, a variant of round-robin scheduling, which we call elevator scheduling, is used to assign to each worker one or more of the top b clusters. This method was chosen after initial analysis showed that the cluster distribution is uneven. After each worker has been assigned its clusters, the worker is responsible for scoring all its clusters data.

The method of scoring items within the top b clusters is the same as when selecting the top clusters. This process of finding top k results is visualized in Figure 3.5. As already mentioned, the distance function is the same as in Blackthorn, but a different

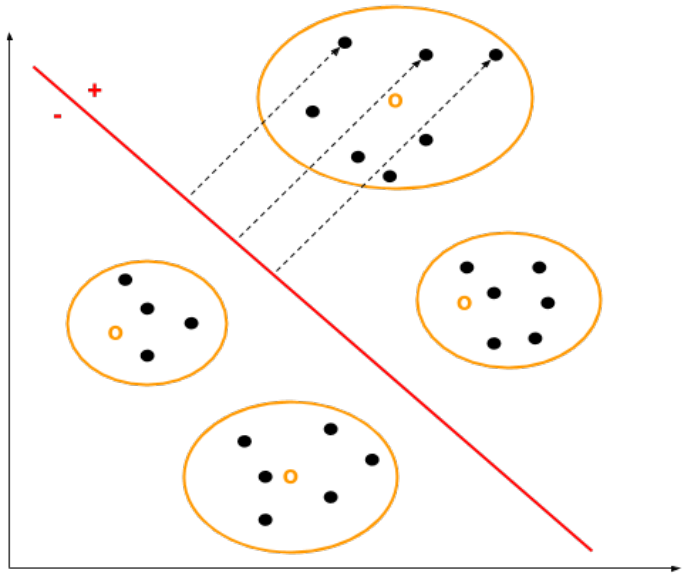

Figure 3.5: Simplified idea of scoring $k = 3$ items in $b = 1$ clusters using a classification boundary. The dotted lines represent the distance to the three farthest (best) items.

method is used for keeping track of the scores and handling already seen items. In Blackthorn Pruning, an ordered list of top k items is dynamically maintained throughout the scoring process for each worker. To keep track of items seen in previous rounds, a single boolean array of size N is used, which informs whether to add an item to the list of top k items.

3.3.3 Merging Top k Results from Both Modalities

We need to combine the k suggested items from both modalities in each worker. The list of top k items in each worker is initialized with the top k items from the visual modality, since the items in the visual modality tend to be the highest scored items. Recall that each modality maintains its list of top items ordered by relevance (highest scored items first). If an item from the textual modality has a higher score than the lowest scored item in the final list, it replaces that item in the list and gets swapped to its correct place in the ordered list. When more than one worker is used all of the lists from the workers are merged into the final top k items, which are then suggested to the user.

3.4 Summary

In this chapter, we have introduced Blackthorn Pruning and described some implementation details regarding the required modification of eCP and Blackthorn. The modification mainly involved adjusting the eCP indexing method to the Ratio-64 representation and adapting the scoring part of Blackthorn to the indexed representation. The performance of Blackthorn Pruning is reported in next chapter.

4 Experiments Evaluation

The largest publicly available multimedia collection is YFCC100M, which consists of 99.206.564 Flickr images [20]. Apart from the images, the data set also consists of associated annotations — title, tags and description — and a range of metadata. The metadata is produced by either the capturing device, the online platform, or the user's geo-location and time stamps. To the best of our knowledge there are no other annotated collections for evaluating multimedia analytics systems, especially not at the scale we are experimenting and therefore we use YFCC100M to evaluate our approach.

We apply top-n based feature selection in both modalities to create the compressed representation meaning that we select at most n top features from the visual concepts and LDA topics with the highest score mentioned in Section 2.2.

As in [2], we further follow the evaluation protocol inspired by the MediaEval Placing Task [21], [22]. We note that although this approach is not designed for estimating capturing location of the images, which is the focus of Placing Task, we find it suitable for general evaluation of analytic performance, and it allows a direct comparison with the results from [2]. From the 2016 edition of the MediaEval Placing Task we create artificial actors by selecting the top-50 world cities represented with the largest number of images. The relevance set of an artificial actor then consists of the images and their associated metadata captured within 1000km from the centre of one city, which is the largest radius used in the Placing Task. A particularly large radius was intentionally selected due to our focus on semantic relevance of the items instead of their exact capturing location. For each actor the evaluation starts by pre-training the interactive classification model using 100 randomly selected relevant images as the positives and another 200 negative examples randomly selected from the collection. In each interaction round the actor is presented with top-25 items selected by the model with 5x5 grid visualization in mind. Then, the items that are part of actor's profile are added to the set of positives and 100 randomly selected items are used as the negatives to train the interactive learning model in the subsequent round. To analyse the system's performance, we measure precision, recall over 10 interaction rounds, recall over time of 50 seconds, and time per interaction round [23].

In the remainder of this section we run a set of experiments seeking to answer the following questions:

1. How is the performance of our approach influenced by the number of clusters b read to produce the final results list presented to the interacting user?

2. What is the influence of the number of workers w on the performance of the system?

3. Is the approach effective in presenting relevant results to the user within time-limited sessions?

4. Does our proposed approach scale to larger collections with up to 1 billion multimedia items?

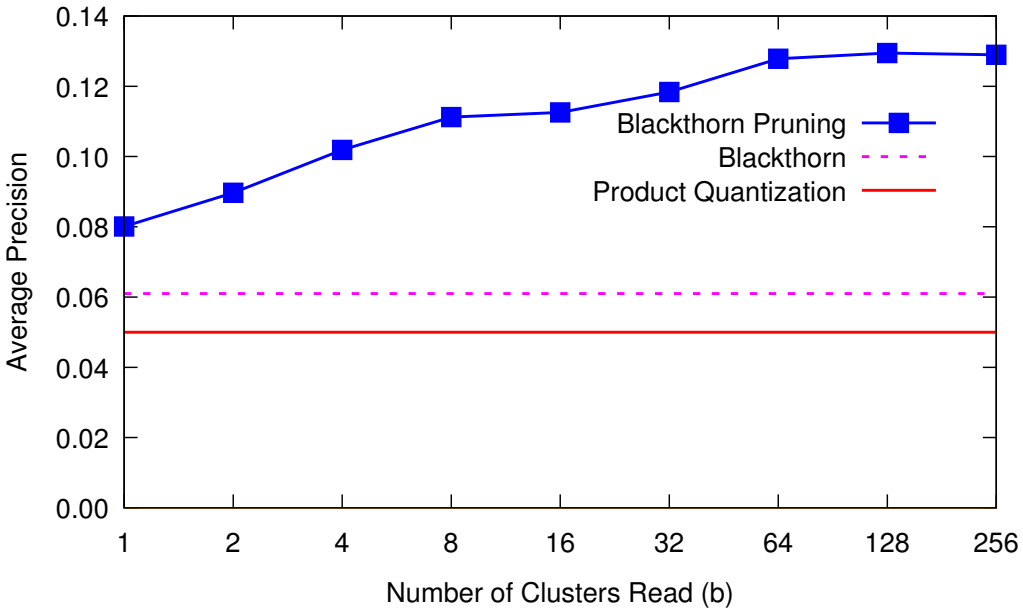

Figure 4.1: Average response time over first 10 rounds of analysis (Result Quality: YFCC100M; varying b; $w = 1$).

4.1 Experimental Setup

All experiments are performed using the DAS-5 server, described in Section 3.1

The index for both visual and textual modality consists of $992,066$ clusters, organized in a modest 3 level deep index hierarchy which gives on average 100 data items per cluster for the 100M dataset. For 1B items the index for both modalities consists of $110,230$ clusters with average cluster size of $9,000$ items of 3 levels.

4.2 Result Quality

In this experiment, we explore the impact of the high-dimensional index on quality. The primary parameter in the scoring process is b, the number of clusters read and scored. Figure 4.1 analyses the impact of b on the result quality in each round of exploration. The x-axis shows how many clusters are read for scoring at each round, ranging from $b = 1$ to $b = 256$, while the y-axis shows the average precision across the first 10 rounds of analysis. For comparison, the figure also shows the average precision for Blackthorn and Product Quantization (reported in [2]).

As Figure 4.1 shows, the quality is improved significantly by focusing scoring on the most relevant clusters. The improvement is seen with only $b = 1$ cluster, but the best results are seen with $b = 128$ clusters, where the quality is more than 2x better than with either Blackthorn original or Product Quantization. Another reason for why Blackthorn Pruning has higher precision than original Blackthorn, already for $b = 1$ cluster, is that a different modality fusion method is used in Blackthorn Pruning, which resulted in a higher precision.

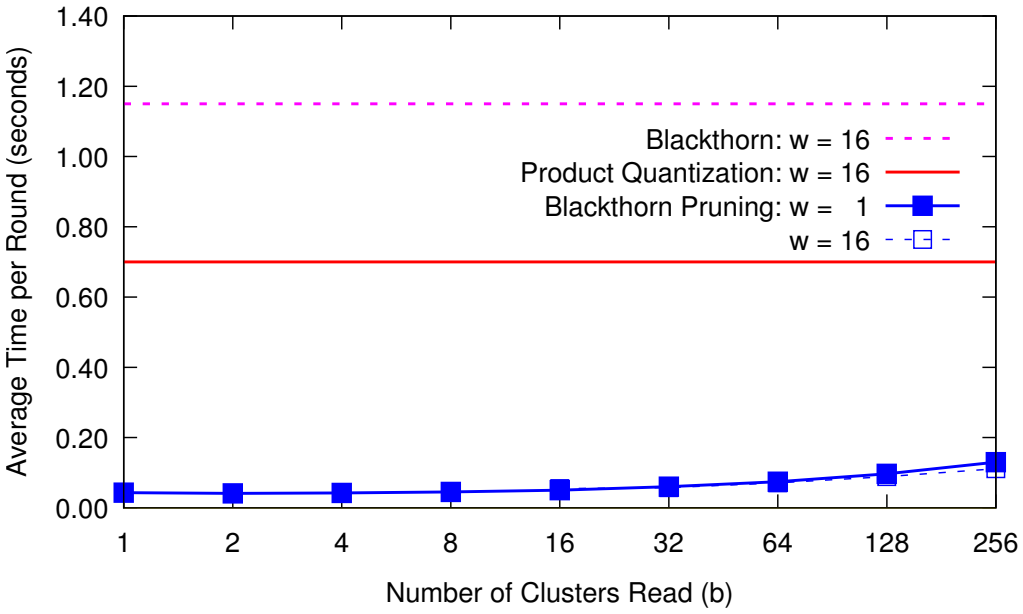

Figure 4.2: Average response time over first 10 rounds of analysis (scoring performance: YFCC100M; varying b; $w = 1, 16$).

4.3 Scoring Performance

In this experiment, we explore the impact of the number of workers on scoring performance. The primary parameter in this experiment is w, the number of workers applied to the scoring process, and we measure this for different values of b, the clusters read and scored. Figure 4.2 analyses the impact of b and w on the response time in each round of exploration. As before, the x-axis shows how many clusters are read for scoring at each round, ranging from $b = 1$ to $b = 256$, while the y-axis shows the average response time across the first 10 rounds of analysis, for $w = 1$ and $w = 16$ (we also measured $w = 2, 4, 8$ but omit these results as they fall between the two extreme values). For comparison, the figure again also shows the response time for Blackthorn (measured by us) and Product Quantization (reported in [2]).

As Figure 4.2 shows, the response time is also improved very significantly by focusing the scoring on the most relevant clusters. For the lowest value of b the response time is very low, but with $b = 16$, the response time starts to slowly increase. For $b = 64$, which showed one of the highest precision, response time is still very low, or 14x smaller than original Blackthorn, making $b = 64$ the preferred configuration. Figure 4.2 furthermore shows that applying more workers gives very limited benefits, but it starts to matter more for higher values of b.

One of the reasons for small benefit of using multiple workers is that Blackthorn Pruning is less than 0.1 second with just one worker so the overhead of adding more workers counteracts the gain from parallelizing the scoring. Another reason is uneven distribution of cluster size in the collection. Figure 4.3 shows how one worker often has significantly more work to do than other workers, making parallelism ineffective for our system. In Section 5.2, we discuss how to combat this.

Figure 4.3: Unbalanced distribution of work between workers (scoring performance: YFCC100M; varying b; $w = 16$).

4.4 Recall Over Time

While high precision at each interaction round is desirable in order to keep the analyst engaged and confident about the system's performance, *recall over time* is an equally important property of an analytic system, as it reflects both cumulative insight gain and the time needed to obtain it. For example, a system yielding high precision (or recall) over very long sessions would be of little practical use for the analyst in an interactive setting. Moreover, in many fields relying on multimedia analytics, such as digital forensics, obtaining high recall in time-limited sessions is even the most important imperative.

Figure 4.4 shows the accumulated recall over time for Blackthorn and Blackthorn Pruning. For Blackthorn Pruning, we set $b = 64$ and use 16 workers, $w = 16$, since Blackthorn uses 16 workers. As the figure shows, Blackthorn Pruning shows more than 50 times better recall. This difference is due to the improved precision and response time.

4.5 Scalability

In this experiment we aim to explore the impact of larger collections on response time. As we do not have access to larger collections than YFCC100M, we follow the lead of [24] and replicate the YFCC100M collection 10 times, resulting in a collection we simply call 1B. The 1B dataset was clustered with around 90 times larger clusters than the YFCC100M dataset as described in Section 4.1. Thus, by using the same workload as before, that is $b = 64$ and $w = 1, 16$, we would assume the time per interaction round to be around 90 times slower than for YFCC100M, or around 6.7 seconds. The reason for why average cluster-size of 9000 was chosen, is that in the original experiments for the YFCC100M dataset, the average cluster-size was chosen

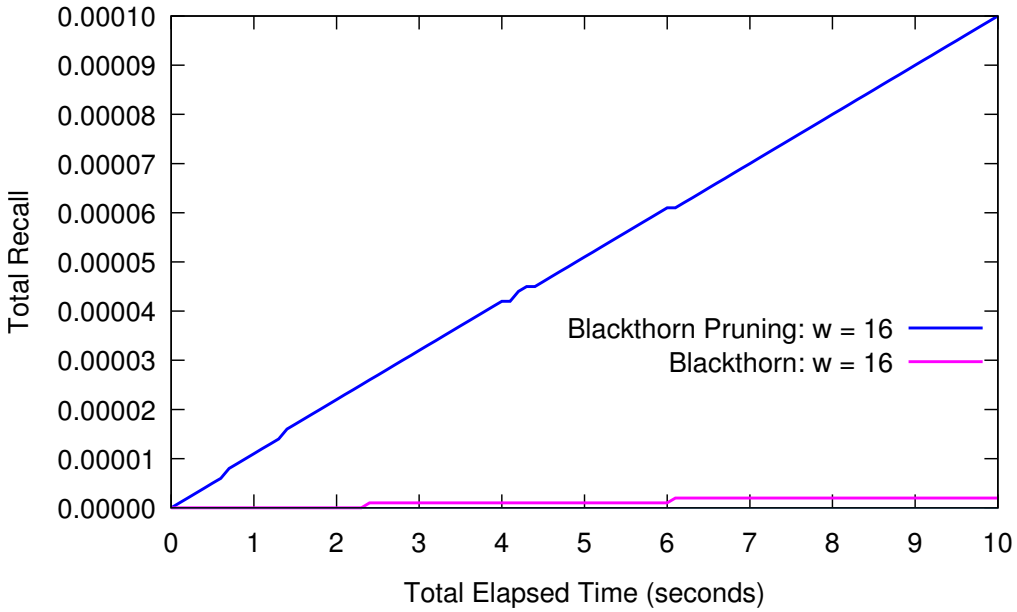

Figure 4.4: Evolution of accumulated recall over 10 seconds (Recall Over Time: YFCC100M; $b = 64$; $w = 16$).

to be 900, and thus for a 10 times larger dataset, we assumed 10 times larger clusters. Since the experiments for YFCC100M were done again with a smaller average cluster-size (of 100), the argument for the choice of size of clusters for 1B does no longer hold. However, time unfortunately did not allow to redo the indexing phase of the 1B dataset. Thus results for the 1B dataset must be interpreted with that in mind.

Results from the experiments showed that one interaction round takes around 4 seconds using $b = 64$ and $w = 1$, which is around 53 times slower than on the YFCC100M. While any quality results must of course be interpreted very carefully on a generated dataset, we observe that precision is higher with the 1B collection than with the YFCC100M collection. The higher precision is explained by the fact that we are exploring more of the 1B dataset than the YFCC100M dataset, since the clusters for 1B are 90 times larger (instead of just 10 times larger, which would result in a similar proportion explored). By doing a small experiment on the 1B collection using only $b = 1$, we get an average response time of around 1 second, and a precision as high as when using $b = 64$. These results support our intuition that the clusters for the 1B collection are too large and we are looking at more data than necessary. Thus by using smaller clusters for the 1B collection (of around 1000), the time per interaction will likely by less than a second, while preserving the precision, making the system still interactive on this large collection. Further exploration on this dataset is needed, and is underway.

4.6 Summary of Results

Table 4.1 summarizes the precision, recall after 10 interaction rounds, and time per interaction round on the YFCC100M dataset for both Blackthorn and the preferred configuration of Blackthorn Pruning ($b = 64$, using 1 and 16 workers). The results clearly reveal that Blackthorn Pruning outperforms Blackthorn with regards to preci-

	Time	**Precision**	**Recall**
Blackthorn Pruning. $w = 1$, $b = 64$	0.074	0.128	$9.11 \cdot 10^{-6}$
Blackthorn Pruning. $w = 16$, $b = 64$	0.074	0.129	$9.29 \cdot 10^{-6}$
Blackthorn. $w = 16$	1.116	0.064	$3.22 \cdot 10^{-6}$

Table 4.1: Summary of the results of experiments on the YFCC100M dataset for Blackthorn and Blackthorn Pruning

sion, recall per 10 interaction round and time per interaction round. This means that the information loss incurred by the high-dimensional index turns into an information gain, making the speed-up of Blackthorn Pruning even more favorable.

5 Discussion

The results presented in this thesis suggest that we are taking a significant step towards bringing large-scale multimedia analytics to the masses. However, further study is needed to completely understand all factors contributing to the large performance improvement, as well as potential failure scenarios. In this chapter, we briefly discuss limitations of Blackthorn Pruning and future work.

5.1 Limitations

One of the limitations of Blackthorn Pruning is the small benefit of adding more workers. Blackthorn Pruning is very efficient using a single worker, and thus adding more workers was not necessarily expected to improve the results much, taking into account the added overhead of more workers. However, the improvement of adding more workers was even less than expected, or negligible.

Parallelization is applied when scoring items of top b clusters, and the top clusters are divided between workers, so each worker gets one or more cluster to score. All the workers must then wait for the worker that takes the longest to score. After analyzing the cluster distribution, we noticed that although most of the clusters are of similar size, there are few clusters that are much larger than the average size clusters. The size of the largest cluster is around 3.5% of the dataset when doing experiments for 100 million items. This explains why the parallelization does not make much difference, as a worker assigned to such large cluster takes a longer time to score its items than all the other workers.

Another limitation of Blackthorn Pruning is the fact that when combining top results from both modalities, the system places more emphasis on visual modality. That method actually proved to be very good for the dataset we tested on and resulted in better precision, but the method should be tested on more datasets to validate its general performance.

The current method of modality fusion also does not account for cases where an item is picked as a top candidate in both modalities. This scenario proved to be very rare but will be accounted for in next version.

The current method of modality fusion also does not account for cases where the same item is picked as a top candidate in both modalities and the item's text score outperforms the score of a top item in the visual modality. This scenario is very rare but code will be fixed to account for this when time allows.

5.2 Future Work

Further study is needed to understand the cluster distribution of the compressed data and to do more experiments with parameters of the indexing phase, such as the number of levels of the index, target size of the clusters, etc. If the cluster distribution is difficult to balance, it is important to improve the scheduling of the workers, so that no single worker is scoring many more items than the others, for example by splitting

up large clusters between many workers. Finally, although results on the 1B dataset are promising, more experiments on a dataset with billion items are necessary to better understand how the system behaves at such a large scale.

6 Conclusion

In this thesis we have presented an approach to large scale interactive multimodal learning, relying on approximate high-dimensional indexing as well as compressed representation and learning in the compressed space. Experiments on the YFCC100M dataset, the largest publicly available multimedia collection, show that our approach yields higher precision than state of the art alternatives, while being significantly faster and yet consuming only a small fraction of the computing power. One interaction round takes less than 0.08 seconds on a single processor core, which is efficient enough to bring large scale multimedia analytics to mobile devices. This high performance is well reflected in increased recall over time, which demonstrates that our approach yields both higher relevance of the items presented to the user and a more timely analysis. Results on the 1B dataset also suggest that our approach stays interactive even on a single computer core, while maintaining similar precision levels obtained on YFCC100M. As a conclusion, results presented in this thesis suggest that we have created the world's most scalable interactive multimodal learning system and are taking a significant step towards bringing large-scale multimedia analytics to the masses.

Bibliography

[1] J. Zahálka and M. Worring, "Towards interactive, intelligent, and integrated multimedia analytics", in *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, Oct. 2014, pp. 3–12. DOI: 10.1109/VAST.2014.7042476.

[2] J. Zahálka, S. Rudinac, B. Þ. Jónsson, D. C. Koelma, and M. Worring, "Blackthorn: Large-scale interactive multimodal learning", *IEEE Transactions on Multimedia*, vol. 20, no. 3, pp. 687–698, Mar. 2018, ISSN: 1520-9210. DOI: 10.1109/TMM.2017.2755986.

[3] G. Þ. Guðmundsson, B. Þ. Jónsson, and L. Amsaleg, "A large-scale performance study of cluster-based high-dimensional indexing", in *Proceedings of the International Workshop on Very-large-scale Multimedia Corpus, Mining and Retrieval*, ser. VLS-MCMR '10, Firenze, Italy: ACM, 2010, pp. 31–36, ISBN: 978-1-4503-0166-4. DOI: 10.1145/1878137.1878145. [Online]. Available: http://doi.acm.org/10.1145/1878137.1878145.

[4] 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, ISSN: 0018-9219. DOI: 10.1109/JPROC.2008.916364.

[5] J. Zahálka, S. Rudinac, and M. Worring, "Interactive multimodal learning for venue recommendation", *IEEE Transactions on Multimedia*, vol. 17, no. 12, pp. 2235–2244, 2015.

[6] A. G. Hauptmann, J. J. Wang, W.-H. Lin, J. Yang, and M. Christel, "Efficient search: The informedia video retrieval system", in *Proceedings of the 2008 International Conference on Content-based Image and Video Retrieval*, ser. CIVR '08, Niagara Falls, Canada: ACM, 2008, pp. 543–544, ISBN: 978-1-60558-070-8. DOI: 10.1145/1386352.1386422. [Online]. Available: http://doi.acm.org/10.1145/1386352.1386422.

[7] C. Snoek, M. Worring, O. de Rooij, K. van de Sande, R. Yan, and A. Hauptmann, "Videolympics: Real-time evaluation of multimedia retrieval systems", *IEEE MM*, vol. 15, no. 1, pp. 86–91, 2008, ISSN: 1070-986X. DOI: 10.1109/MMUL.2008.21.

[8] K. Schoeffmann, "A user-centric media retrieval competition: The Video Browser Showdown 2012-2014", *IEEE MM*, vol. 21, no. 4, pp. 8–13, 2014, ISSN: 1070-986X. DOI: 10.1109/MMUL.2014.56.

[9] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, 2011.

[10] 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", *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, Dec. 2015, ISSN: 1573-1405. DOI: 10.1007/s11263-015-0816-y. [Online]. Available: https://doi.org/10.1007/s11263-015-0816-y.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions", in *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, Jun. 2015, pp. 1–9. DOI: 10.1109/CVPR.2015.7298594.

[12] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora", English, in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, http://is.muni.cz/publication/884893/en, Valletta, Malta: ELRA, May 2010, pp. 45–50.

[13] C. G. M. Snoek, M. Worring, and A. W. M. Smeulders, "Early versus late fusion in semantic video analysis", in *Proceedings of the 13th Annual ACM International Conference on Multimedia*, ser. MULTIMEDIA '05, Hilton, Singapore: ACM, 2005, pp. 399–402, ISBN: 1-59593-044-2. DOI: 10.1145/1101149.1101236. [Online]. Available: http://doi.acm.org/10.1145/1101149.1101236.

[14] R. Tavenard, H. Jégou, and L. Amsaleg, "Balancing clusters to reduce response time variability in large scale image search", in *International Workshop on Content-Based Multimedia Indexing*, Madrid, Spain, Jun. 2011.

[15] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases", in *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2008.

[16] G. Þ. Guðmundsson, L. Amsaleg, and B. Þ. Jónsson, "Impact of storage technology on the efficiency of cluster-based high-dimensional index creation", in *Proceedings of the 17th International Conference on Database Systems for Advanced Applications*, ser. DASFAA'12, Busan, South Korea: Springer-Verlag, 2012, pp. 53–64, ISBN: 978-3-642-29022-0. DOI: 10.1007/978-3-642-29023-7_6. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-29023-7_6.

[17] D. Moise, D. Shestakov, G. Þ. Guðmundsson, and L. Amsaleg, "Indexing and searching 100m images with map-reduce", in *Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval*, ser. ICMR '13, Dallas, Texas, USA: ACM, 2013, pp. 17–24, ISBN: 978-1-4503-2033-7. DOI: 10.1145/2461466.2461470. [Online]. Available: http://doi.acm.org/10.1145/2461466.2461470.

[18] G. Þ. Guðmundsson, L. Amsaleg, B. Þ. Jónsson, and M. J. Franklin, "Towards engineering a web-scale multimedia service: A case study using Spark", in *Proceedings of the 8th ACM on Multimedia Systems Conference*, ser. MMSys'17, Taipei, Taiwan: ACM, 2017, pp. 1–12, ISBN: 978-1-4503-5002-0. DOI: 10.1145/3083187.3083200. [Online]. Available: http://doi.acm.org/10.1145/3083187.3083200.

[19] H. Bal, D. Epema, C. D. Laat, R. V. Nieuwpoort, J. Romein, F. Seinstra, C. Snoek, and H. Wijshoff, "A medium-scale distributed system for computer science research: Infrastructure for the long term", *Computer*, vol. 49, no. 5, pp. 54–63, 2016. DOI: 10.1109/mc.2016.127.

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

[21] 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 *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, ser. ICMR '11, Trento, Italy: ACM, 2011, 51:1–51:8, ISBN: 978-1-4503-0336-1. DOI: 10.1145/1991996.1992047. [Online]. Available: http://doi.acm.org/10.1145/1991996.1992047.

[22] J. Choi, C. Hauff, O. van Laere, and B. Thomee, "The placing task at mediaeval 2015", in *Proceedings of the MediaEval 2015 Workshop*, M. Larson, B. Ionescu, M. Sjoberg, X. Anguera, J. Poignant, M. Riegler, M. Eskevich, C. Hauff, R. Sutcliffe, G. Jones, Y. Yang, M. Soleymani, and S. Papadopoulos, Eds. CEUR, 2015, pp. 1–2.

[23] J. Zahálka, S. Rudinac, and M. Worring, "Analytic quality: Evaluation of performance and insight in multimedia collection analysis", in *Proceedings of the 23rd ACM International Conference on Multimedia*, ser. MM '15, Brisbane, Australia: ACM, 2015, pp. 231–240, ISBN: 978-1-4503-3459-4. DOI: 10.1145/2733373.2806279. [Online]. Available: http://doi.acm.org/10.1145/2733373.2806279.