QuantumCLEF 2024: Overview of the Quantum Computing Challenge for Information Retrieval and Recommender Systems at CLEF

Andrea Pasin[1,*], Maurizio Ferrari Dacrema[2], Paolo Cremonesi[2] and Nicola Ferro[2]

[1] *University of Padua, Padua, Italy*
[2] *Politecnico di Milano, Milano, Italy*

Abstract

The emerging field of *Quantum Computing (QC)* in computational science is attracting significant research interest due to its potential for groundbreaking applications. In fact, it is believed that QC could potentially revolutionize the way we solve very complex problems by significantly decreasing the time required to solve them. Even though QC is still in its early stages of development, it is already possible to tackle some problems using quantum computers and, thus, begin to see its potential. Therefore, the aim of the QuantumCLEF lab is to raise awareness about QC and to develop and evaluate new QC algorithms to solve challenges that are usually faced when implementing *Information Retrieval (IR)* and *Recommender Systems (RS)* systems. Furthermore, this lab represents a good opportunity to engage with QC technologies, which are typically not easily accessible due to their early development stage.

In this work, we present an overview of the first edition of QuantumCLEF, a lab that focuses on the application of *Quantum Annealing (QA)*, a specific QC paradigm, to solve two tasks: Feature Selection for IR and RS systems, and Clustering for IR systems. There were a total of 26 teams who registered for this lab, and eventually, 7 teams successfully submitted their runs following the lab guidelines. Due to the novelty of the topics, participants were provided with many examples and comprehensive materials to help them understand how QA works and how to program quantum annealers.

Keywords

Quantum Computing, Quantum Annealing, CLEF, Information Retrieval, Recommender Systems

1. Introduction

Even though *Information Retrieval (IR)* and *Recommender Systems (RS)* systems have been studied and improved for several years, they still face significant challenges. The ever-growing volume of data and the need for computationally intensive methods to analyze it represent a complex task for these systems.

To address these challenges, researchers are now investigating *Quantum Computing (QC)*, an emerging computing paradigm that has the potential to revolutionize the way we currently solve problems. QC is not simply a new technology that can be employed instead of traditional hardware; it represents a paradigm shift that allows us to view and solve problems from a new perspective by exploiting quantum physics principles. Unlike classical computing, which processes information in a binary manner (i.e., each bit can be either 0 or 1), QC uses quantum bits, or qubits, which can exist in multiple states simultaneously due to the superposition principle. Additionally, qubits can be entangled, meaning the state of one qubit can depend on the state of another, no matter the distance between them.

This allows quantum computers to theoretically explore exponentially larger problem spaces compared to traditional computers for some specific problems (i.e., problems for which the quantum physics principles can be correctly exploited to find a solution). This fundamental shift could lead to significant

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France

[*] Corresponding author.

andrea.pasin.1@phd.unipd.it (A. Pasin); maurizio.ferrari@polimi.it (M. Ferrari Dacrema); paolo.cremonesi@polimi.it (P. Cremonesi); nicola.ferro@unipd.it (N. Ferro)
https://www.dei.unipd.it/~pasinandre/ (A. Pasin); https://www.dei.unipd.it/~ferro/ (N. Ferro)
0009-0007-5193-0741 (A. Pasin); 0000-0001-7103-2788 (M. Ferrari Dacrema); 0000-0002-1253-8081 (P. Cremonesi); 0000-0001-9219-6239 (N. Ferro)

advancements in the efficiency and capability of IR and RS systems, allowing them to handle complex computations more efficiently. As a result, the integration of QC into these systems could unlock new possibilities and provide new performing solutions, especially once QC technologies will be mature enough. In fact, even though quantum computers have started to become more robust and accessible, QC is still in its infancy and there are several limitations yet to overcome, most of which concerning the hardware. In fact, qubits are very delicate and must be completely isolated from the environment since any interferences or noises (e.g., electromagnetic interferences, thermal fluctuations) could impact their state, thus breaking the computation. On the other hand, traditional systems have been developed for decades and they represent more robust alternatives.

In this exciting and innovative context, it is natural to wonder whether it is possible to apply QC to solve some of the complex tasks that are faced by IR and RS systems. To explore the QC's potential, we decided to start a new CLEF lab called QuantumCLEF [1, 2], focusing on the study, development, and evaluation of QC algorithms for IR and RS. This lab has four main goals:

- develop new QC algorithms for IR and RS, and evaluate their performance by comparing efficiency and effectiveness with traditional approaches;
- gather all resources and data to allow future researchers to compare their results with those achieved during the lab;
- provide participants with comprehensive materials to learn more about QC and offer them access to real quantum computers, which are not easily accessible to the public yet;
- raise awareness of QC's potential and foster a new research community around this emerging field.

In this paper, we present an overview of the first edition of QuantumCLEF held in 2024. This edition focused on the use of *Quantum Annealing (QA)*, a specific QC paradigm designed to tackle optimization problems. Participants were granted access to cutting-edge QA devices (quantum annealers) produced by D-Wave, one of the leading companies in this sector.

The QA paradigm is more accessible and easier to understand compared to the Universal Gate-Based paradigm. Additionally, D-Wave provides several tools and libraries that facilitate programming quantum annealers, allowing participants to engage with these advanced devices without needing a profound knowledge of the underlying quantum physics principles governing the quantum computers' behaviour. This resulted in a more approachable technology for a broader range of researchers, allowing participants to only focus on the development and testing of their approaches. In this way, we managed to show the applicability and feasibility of QA to solve real problems.

This QuantumCLEF edition was composed of two main tasks:

- **Task 1**: Feature Selection for IR and RS;
- **Task 2**: Clustering for IR.

Participants were asked to develop their own algorithms to solve challenges using both QA and *Simulated Annealing (SA)*. SA is a well-known optimization approach similar to QA but suitable for classical devices. Given the novelty of the topics, comprehensive materials—including videos, slides, and examples—were provided to participants to lower the entry barrier and help them understand how QA works and how to program quantum annealers.

To support this initiative, an ad-hoc infrastructure was created to grant participants access to real quantum annealers. This infrastructure was designed to streamline the workflow and enhance reproducibility of the experiments. A total of 26 teams registered for our tasks, with 7 teams actively participating and submitting their runs. Specifically, 6 teams successfully submitted their runs for Task 1, while 1 team managed to submit for Task 2.

The results demonstrate that approaches using QA or hybrid methods are as effective as those using SA and traditional approaches while generally being more efficient. This shows that QA is indeed a feasible and effective approach when tackling complex optimization problems within the realms of IR, RS, and possibly many other research fields.

The paper is organized as follows: Section 2 discusses related works; Section 3 presents the tasks of the QuantumCLEF 2024 lab while Section 4.1 introduces the lab's setup and the design and implementation of our ad-hoc infrastructure; Section 5 shows and discusses the results achieved by the participants; finally, Section 6 draws some conclusions and outlooks some future work.

2. Related Works

2.1. Background on Quantum and Simulated Annealing

We present here a brief introduction to QA and *Simulated Annealing (SA)*, a traditional optimization algorithm that does not leverage quantum technologies.

2.1.1. Quantum Annealing.

QA is a QC paradigm that is based on special-purpose devices known as quantum annealers to tackle optimization problems with a specific structure. The fundamental concept of a quantum annealer is to represent a problem as the energy of a physical system. It then leverages quantum-mechanical phenomena, such as superposition and entanglement, to let the system find a state of minimal energy, which corresponds to the solution of the original problem.

To use quantum annealers, one needs to formulate the optimization problem as a minimization one using the *Quadratic Unconstrained Binary Optimization (QUBO)* formulation [3], a well-known optimization technique. QUBO is defined as:

$$\min \quad y = x^T Q x \tag{1}$$

where x is a vector of binary decision variables, and Q is a matrix of constant values representing the problem we wish to solve. Then, a further step called *minor embedding* is required to map the general mathematical formulation into the physical quantum annealer hardware, accounting for the limited number of qubits and the physical connections between them. While traditional computers have CPUs, each quantum annealer has a *Quantum Processing Unit (QPU)* with its own architecture, which can be seen as a graph: each vertex represents a qubit, and each edge represents an interaction between two qubits. Therefore, minor embedding involves selecting which physical qubits will represent the decision variables. If the QUBO problem does not fit directly in the QPU, for example because a decision variable is connected to more variables than the available physical connections between qubits, multiple connected qubits will be used to represent one decision variable and the connections to the other variables will be split between them. Consequently, the number of qubits required to solve a problem on a quantum annealer may be much higher than the number of its decision variables. Minor embedding is a complex task and a NP-hard problem, which can be addressed relying on some heuristic methods [4]. If the problem does not fit on the QPU, D-Wave provides *Hybrid (H)* approaches that are able to automatically handle large problems using intelligent techniques to split them and solve them using both traditional methods and QA methods. By splitting problems into smaller sub-problems it will be possible to make them fit inside the QPU of quantum annealers.

Occasionally, it might be necessary to add constraints to the problems. This can be accomplished using penalties $\mathrm{P}(x)$ [5], which penalize solutions that do not satisfy the specified constraints. These penalties are then incorporated to the original cost function y to achieve the final formulation as follows:

$$\min \quad \mathrm{C}(x) = y + \mathrm{P}(x)\,. \tag{2}$$

Penalties can be controlled through hyperparameters to manage their influence with respect to the given formulation.

To sum up, using a quantum annealer requires several stages [5]:

1. **Formulation**: find a way to express the desired algorithm as an optimization problem by leveraging the QUBO framework and compute the actual QUBO matrix Q;
2. **Embedding**: generate the minor embedding of the QUBO for the quantum annealer hardware;
3. **Data Transfer**: transfer the problem and the embedding on the global network to the data center that hosts the quantum annealer;
4. **Annealing**: run the quantum annealer itself. This phase is composed of several stages such as programming the QPU, sampling a solution, and then reading the solution. This is an inherently stochastic process. Therefore, it is usually run a large number of times (hundreds) in which several samples are returned, each one resembling a possible solution to the considered problem. The solutions must then be checked for their feasibility, and then the best one among them (i.e., the optimal one according to the objective function) is usually considered the final solution to the submitted problem.

Generally, once a QUBO problem has been embedded and sent to the quantum annealer, it can be solved in a few milliseconds.

2.1.2. Simulated Annealing.

SA is a consolidated meta-heuristic that can be run on traditional hardware [6, 7]. It is a probabilistic algorithm that can be used to find the global minimum of a given cost function, even in the presence of many local minima. The algorithm operates through an iterative process that begins with an initial solution and attempts to improve it by randomly perturbing it. The cost function can be represented by the QUBO problem formulation, similar to what would be used for QA. SA is inspired by the annealing process in metallurgy, a technique that involves heating and gradually cooling a material to alter its physical properties, which also translates into minimizing the system's energy. In SA, there is no minor embedding phase since the problem is directly solved on a traditional machine.

We underline that SA is an optimization algorithm different from QA, it is not a simulation of QA on traditional hardware, and, therefore these two algorithms are not equivalent. However, SA can be used for benchmarking purposes to show how well QA performs with respect to a traditional hardware counterpart.

The access to quantum annealers in QuantumCLEF is limited to ensure a fair distribution of resources. Therefore, SA can also be used to conduct initial experiments to assess the feasibility of a QUBO formulation without affecting the available QC quota.

2.2. Related Challenges

In the context of CLEF, there have not been other challenges involving the application and evaluation of QC. However, since QC technologies are starting to become more available and robust, it is necessary to raise awareness about their potential and to learn how these technologies can be used to possibly improve the current state-of-the-art IR and RS systems.

Outside CLEF, we are not aware of other challenges or shared tasks that have been done in the past involving the use of QC. There are some other challenges starting off this year offered by big-tech companies such as IBM[1] and Google[2]. These challenges involve the development of QC algorithms which will be executed on quantum computers to solve some practical real-world challenges. There has also been a Quantum Computing challenge in 2016 organized by Microsoft[3], which however used simulators for Language-Integrated Quantum Operations and not real quantum computers.

[1] https://challenges.quantum.ibm.com/2024

[2] https://www.xprize.org/prizes/qc-apps

[3] https://www.microsoft.com/en-us/research/academic-program/microsoft-quantum-challenge/challenge/

3. Tasks

QuantumCLEF 2024, initially introduced in a paper at CLEF 2023 [1], proposes two different tasks involving computationally intensive problems that are closely related to the Information Access field: Feature Selection and Clustering. The main objectives for each task are:

- identifying one or more possible QUBO formulations of the problem;
- evaluating the QA approach compared to a corresponding traditional approach to assess both its efficiency and its effectiveness.

For each task, we provided Jupyter Notebooks that served as starting points for the participants to learn how to program quantum annealers and successfully carry out the tasks following the submission guidelines. Moreover, we provided the slides that were presented during the ECIR Tutorial [8] covering the fundamental concepts of QC and QA. We also streamed and recorded a video tutorial[4] about the usage of our infrastructure and the notebooks available to the participants.

For both tasks, participants were asked to submit their runs using both QA and SA. In this way, it is possible to compare the efficiency and effectiveness of these two similar optimization techniques that employ quantum annealers and traditional hardware respectively.

3.1. Task 1 - Quantum Feature Selection

This task focuses on reformulating the well-known *NP-Hard* Feature Selection problem to make it solvable using a quantum annealer in a similar way to what has been successfully done in previous works [9, 10]. Given the NP-Hard nature of this problem, conventional optimization techniques often face significant challenges in terms of scalability and efficiency.

3.1.1. Objectives.

Feature Selection is a widespread problem for both IR and RS. It involves identifying a subset of the available features (e.g., the most informative or least noisy ones) to train a learning model. This problem has significant implications since optimizing the selection of features can greatly improve the performance of learning models by reducing the dimensionality of input data. In many IR and RS systems, the optimization of these models is essential for achieving better accuracy and efficiency.

In this task, our goal is to explore whether QA can be applied to solve the Feature Selection problem more efficiently and effectively. QA has the potential to explore large problem spaces in a short amount of time due to its quantum-mechanical properties. Through this task, we hope to gain insights into the practical advantages and limitations of using QA for Feature Selection, exploring new effective and efficient optimization strategies. Eventually, we aim to determine if QA can provide comparative or superior solutions to the Feature Selection problem compared to traditional methods.

3.1.2. Sub-tasks.

Task 1 is divided into two sub-tasks:

- **Task 1A**: Feature Selection for IR. This task involves selecting the optimal subset of features using QA and SA that will be used to train a LambdaMART [11] model according to a Learning-To-Rank framework;
- **Task 1B**: Feature Selection for RS. This task involves selecting the optimal subset of features using QA and SA that will be used to train a kNN recommendation system model. The item-item similarity is computed with cosine on the feature vectors, a shrinkage of 5 is added to the denominator and the number of selected neighbors for each item is 100.

[4] https://www.youtube.com/watch?v=fKrnaJn40Kk/

3.1.3. Datasets.

For Task 1A, we decided to employ the famous MQ2007 [12] and the Istella S-LETOR [13] datasets. MQ2007 represents an easier challenge since it has 46 features, allowing direct embedding of the problem formulations inside the QPU of quantum annealers. Istella instead has 220 features and it is impossible to embed problem formulations directly, thus requiring some further processing steps for the participants to fit the problem into the physical QPU hardware.

For Task 1B instead, we decided to employ a custom dataset of music recommendations containing 1.9 thousand users and 18 thousand items. The dataset contains both collaborative data, with 92 thousand implicit user-item interactions, as well as two different sets of item features that are derived from item descriptions and user-provided tags, called Item Content Matrix (ICM). The small set, ICM_150, includes 150 features and can be embedded directly on the QPU with small adjustments, the large set, ICM_500, has 500 features and requires significant pruning to fit in the QPU or the use of Hybrid methods. Both sets of features contain noisy and redundant features.

3.1.4. Evaluation Measures.

The official evaluation measure for both Task 1A and Task 1B is nDCG@10.

3.1.5. Baseline.

For sub-task 1A the baseline is a Feature Selection model that uses a Recursive Feature Elimination approach paired with a Linear Regression model to select the most relevant subset of features.

For sub-task 1B the baseline is a kNN recommendation system model that uses all the available features. The hyperparameters are the same used for the model computed on the selected features, i.e., the item-item similarity is computed with cosine adding a shrink term of 5 to the denominator, and the number of neighbors is 100.

3.1.6. Runs Format.

Participants in both tasks 1A and 1B can submit a maximum of 5 runs per dataset using QA or Hybrid methods and a maximum of 5 runs using SA. Each run that uses QA or Hybrid methods should correspond to a run that employs SA. In this way, it is possible to make a fair comparison between them.

The results of the run must be a text file which lists the features that were selected, one per line. The discarded features are not reported in the run file. Furthermore, the last line must report the list of IDs associated with the problems solved using QA, SA, or Hybrid to obtain the final subset of features by the considered approach.

Each run file must be left in each team's workspace in a specific directory called */config/workspace/submissions*, which is already available.

The submission file name should comply with the format
[Task]_[Dataset]_[Method]_[Groupname]_[SubmissionID].txt, where:

- **[Task]**: it should be either *1A* or *1B* based on the task the submission refers to;
- **[Dataset]**: it should be either *MQ2007*, *Istella*, *150_ICM* or *500_ICM* based on the dataset used;
- **[Method]**: it should be either *QA* or *SA* based on the method used;
- **[Groupname]**: the team name;
- **[SubmissionID]**: a custom submission ID that must be the same for the submissions using the same algorithm but performed with different methods (e.g., QA or SA).

3.2. Task 2 - Quantum Clustering

This task focuses on formulating the Clustering problem to solve it using a quantum annealer. Clustering involves organizing items into groups based on their similarities so that similar items are grouped together while dissimilar items are assigned to different groups. This process plays a crucial role in various fields and can be very important in the context of Dense Retrieval approaches in IR.

3.2.1. Objectives.

Clustering is a relevant problem for IR and RS since it can be helpful for organizing large collections, facilitating users to explore a collection, and providing similar search results to a given query. Moreover, it can be beneficial to segment users according to their interests or build user models with the cluster centroids [14] speeding up the runtime of the system or its effectiveness for users with limited data.

This task is more focused on the IR field and is applied in a document retrieval scenario where documents are represented as embeddings derived from Transformer models. Each document can be seen as a vector in a high-dimensional space and it is possible to cluster points based on their distances, which can be interpreted as a dissimilarity function: the farther apart two vectors are, the more different the corresponding documents are likely to be. In this task, participants should apply QA and SA to cluster documents into 10, 25, and 50 clusters. Participants must report the found centroids and the documents associated with each centroid.

Clustering documents offers the advantage of reducing search time by matching an input query to the most similar centroid and retrieving documents only from that cluster, rather than searching the entire document collection.

Clustering fits very well with a QUBO formulation and various methods have already been proposed [15, 16, 17]. Most of these methods involve the usage of one variable per document, thus making it very hard to consider large datasets due to the limited number of physical qubits and interconnections between them. There are ways to overcome this issue, such as by applying a coarsening or a hierarchical approach. By tackling this task, participants explore novel approaches to document clustering that leverage QC's potential for handling complex optimization problems.

3.2.2. Datasets.

For this task, we considered a custom split of the ANTIQUE [18] dataset containing 6486 documents, 200 queries, and manual relevance judgments. Each document and each query have been transformed into a corresponding embedding with the pre-trained **all-mpnet-base-v2** model[5]. The queries are divided into 50 for the Training Dataset and 150 for the Test Dataset. Figure 1 shows a t-SNE visualization [19] of the ANTIQUE dataset with also the 50 training queries provided to the participants.

3.2.3. Evaluation Measures.

The official evaluation measures for Task 2 are:

- the Davies-Bouldin Index to measure the overall cluster quality without considering the document retrieval phase;
- nDCG@10 to measure the retrieval effectiveness based on the clusters found.

3.2.4. Baseline.

For this task, the baseline is a traditional k-Medoids approach using the cosine distance as a distance function.

[5] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

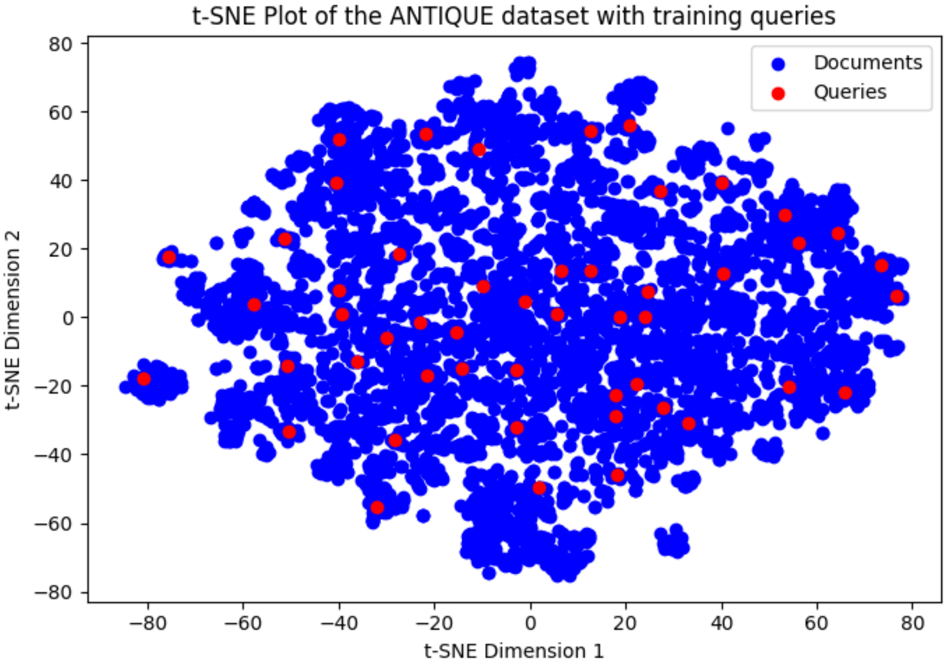


Figure 1: The t-SNE plot of the documents and training queries in the ANTIQUE dataset.

3.2.5. Runs Format.

Participants in task 2 can submit a maximum of 5 runs for each number of clusters (i.e., 10, 25, 50) using QA or Hybrid methods and a maximum of 5 runs using SA. Each run that uses QA or Hybrid methods should correspond to a run that employs SA. In this way, it is possible to make a fair comparison between them.

The run file must be a text file (JSON formatted) with a list of 10, 25, and 50 vectors that represent the final centroids achieved through their clustering algorithm. Each centroid should also be followed by the list of documents that belong to the given cluster. Furthermore, the last line must report the list of IDs associated with the problems solved using QA, SA, or Hybrid to obtain the final clusters by the considered approach.

Each run file must be left in each team's workspace in a specific directory called */config/workspace/-submissions*, which is already available.

The submission file name should comply with the format
[Centroids]_[Method]_[Groupname]_[SubmissionID].txt, where:

- **[Centroids]**: it should be either 10, 25, or 50 based on the number of centroids;
- **[Method]**: it should be either *QA* or *SA* based on the method used;
- **[Groupname]**: the team name;
- **[SubmissionID]**: a custom submission ID that must be the same for the submissions using the same algorithm but performed with different methods (e.g., QA or SA).

4. Lab Setup

In this section, we detail the infrastructure that was specifically created to carry out this lab and we present the guidelines the participants had to comply with to submit their runs.

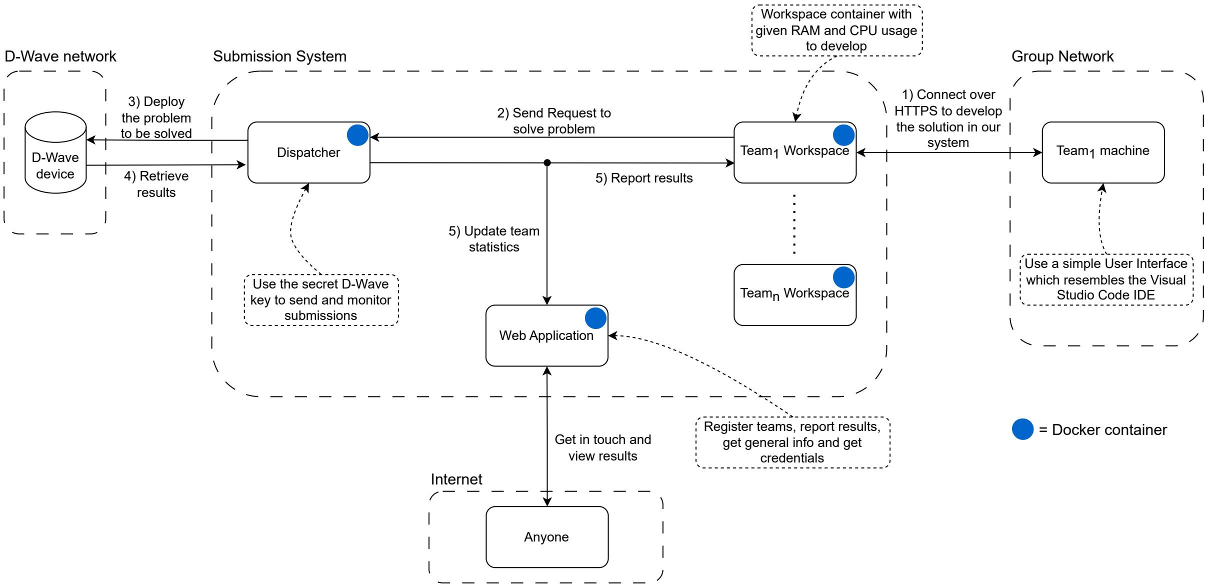

Workspace container with given RAM and CPU usage to develop
D-Wave network
Submission System
Group Network
D-Wave device
3) Deploy the problem to be solved
4) Retrieve results
Dispatcher
2) Send Request to solve problem
Team$_1$ Workspace
1) Connect over HTTPS to develop the solution in our system
Team$_1$ machine
5) Report results
5) Update team statistics
Team$_n$ Workspace
Use the secret D-Wave key to send and monitor submissions
Web Application
Use a simple User Interface which resembles the Visual Studio Code IDE
Get in touch and view results
Register teams, report results, get general info and get credentials
● = Docker container
Internet
Anyone

Figure 2: High-level representation of the infrastructure.

Table 1
The hardware resources corresponding to the AWS EC2 instance and to the participants' workspaces.

-	CPU	RAM	Hard Drive
Infrastructure	32 cores	128 GB RAM	1 TB HDD
Workspace	1200 millicores	10 GB RAM	20 GB HDD

4.1. Infrastructure

Having access to quantum annealers is not straightforward. In fact, D-Wave enforces some policies on the usage of these devices by setting monthly timing quotas to submit and solve problems on their devices. Users are assigned API keys to monitor and regulate access and usage to ensure a fair distribution of quantum resources.

Since it is not possible to disclose our API key to the participants, we decided to build our own infrastructure that allows participants to use quantum annealers without knowing our API key and without needing to stipulate any agreements with D-Wave to obtain their own API keys.

Furthermore, to measure efficiency participants must use the same computing hardware. To this end, our infrastructure provides all the participants with corresponding workspaces located in an AWS server. All workspaces have identical computational resources in terms of CPU and RAM, thus ensuring also easy reproducibility since all computations are performed under the same conditions.

Finally, we wanted to create a workflow that was as easy as possible. To this end, participants can access our infrastructure directly from the Web through a user-friendly interface. This interface not only allows them to monitor their quotas but also allows them to develop and execute their code directly from their web browsers, without having to worry about installing anything on their machines or dealing with command-line tools.

This infrastructure has been implemented using Docker images orchestrated through Kubernetes. It is made up of several components that are interconnected together to provide both organizers and participants easy access to the needed resources, see Figure 2. All problems submitted by the participants were saved in a database to monitor their quotas and to gather data to draw statistics about the lab.

The final infrastructure was deployed on a *m6a.8xlarge* AWS EC2 instance equipped with an AMD EPYC 7R13 processor. Table 1 reports the specifications of the hardware resources corresponding to that instance and to each team's workspace. All participants were given the same monthly quota to use quantum resources. Table 2 reports the monthly quotas according to the two tasks.

Table 2
The monthly quotas to use quantum resources according to the tasks.

Task	March	April	May
Task 1: Feature Selection	30 seconds	30 seconds	50 seconds
Task 2: Clustering	50 seconds	50 seconds	150 seconds

Table 3
The teams who participated and submitted at QuantumCLEF 2024.

Team	Affiliation	Country
BIT.UA	IEETA/DETI, LASI, University of Aveiro	Portugal
CRUISE	RMIT University	Australia
NICA	Iran University of Science and Technology, Departement of Computer Engineering	Iran, Islamic Republic Of
OWS	Friedrich Schiller Universität Jena	Germany
qIIMAS	Universidad Nacional Autonoma de Mexico	Mexico
QTB	Universidad Tecnologica de Bolivar	Colombia
shm2024	Madras Christian College, Chennai	India

4.2. General guidelines

Each team has access to its personal area inside our infrastructure with the credentials that have been provided to them. All runs must be executed within the designated workspaces that have been created and assigned to each one of the participating teams, thus ensuring a fair comparison and easy reproducibility.

All participants cannot exceed their given quotas (see Table 2) to execute problems on quantum devices to ensure a fair distribution of resources. Each team can monitor their quota utilization through a dashboard that is constantly being automatically updated, reporting usages of the different methods (i.e., QA, H, and SA) and some general statistics.

All participants' runs must follow the file formats that are already described in Section 3.1.6 and 3.2.5 to allow us running our evaluation tools smoothly.

Participants have also been asked to upload their files on their dedicated Bitbucket git repositories to enhance transparency and reproducibility. Each repository has been created by us inside a Bitbucket project[6]. Their repositories have been kept private through the challenge but are now public. In this way, the participants' approaches are now easily accessible for further analysis.

5. Results

In this Section, we present the results achieved by the participants and we discuss their approaches. Out of the 26 registered teams, 7 teams managed to upload some final runs. In total, the number of runs is 66 considering both SA, QA, and H (H was introduced in Section 2.1.1). Table 3 reports the 7 teams that correctly participated and submitted some final runs. Table 4 shows the number of runs submitted for each task and subtask. As it is possible to see, there has been a higher number of runs for Task 1. This may be because Task 1 was likely the easiest to address, both due to its lower complexity (e.g., datasets with a relatively low number of features) and the availability of provided code examples.

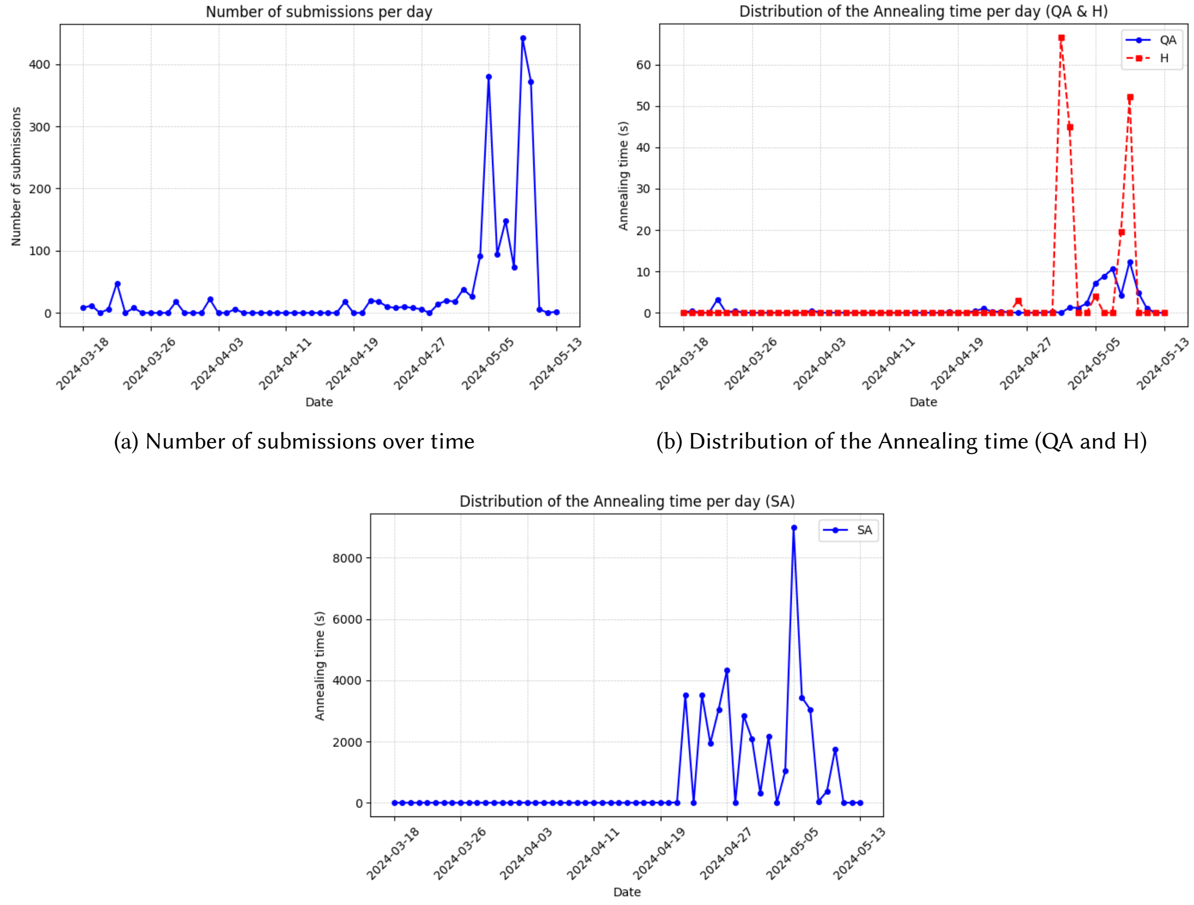
In total, throughout the entire lab participants have submitted 976 problems. Specifically, 758 of them were solved with SA, while 199 were solved using QA and 18 with the H method. The total execution time of SA has been almost 12 hours while the total QA and H execution time has been roughly 4 minutes. The *Annealing time* in this whole Section refers to the *Annealing* phase as described in Section 2.1.1, therefore it includes the time required to program the QPU, sampling, and reading the result. The embedding time and network latencies are not taken into account and are left to be considered

[6] https://bitbucket.org/eval-labs/workspace/projects/QCLEF24

Table 4
The breakdown of the runs submitted by the participating teams for each task and subtask.

Team	Task 1		Task 2
	A	B	
BIT.UA	9	-	-
CRUISE	-	14	-
NICA	4	1	-
OWS	12	-	-
qIIMAS	-	-	12
QTB	4		-
shm2024	10	-	-
Total	39	15	12

for possible future editions of the QuantumCLEF lab. Figure 3 illustrates the temporal distribution of participants' submissions. In particular, it reveals a significant spike in submissions during the final days of the Lab, thus putting the infrastructure under a substantial workload during that period. This shows that the infrastructure was put under heavy load in the last period. In addition, we can see the importance of allocating higher quotas in the final month, as participants tend to concentrate their efforts and finalize their submissions during this time.



(a) Number of submissions over time

(b) Distribution of the Annealing time (QA and H)

(c) Distribution of Annealing time (SA)

Figure 3: The distribution of the participating teams' submissions over time, considering also the Annealing time used per day.

5.1. Task 1A

Here we present the results achieved by the teams participating in task 1A.

5.1.1. MQ2007 dataset.

As it is possible to see in Table 5, teams considered different numbers of features in their submissions. In general, we can observe that most of the submissions achieve similar nDCG@10 values when considering a number of features that lies between 10 and 25. In fact, Figure 4 shows that for these runs the Tukey HSD test performed after the Two-Way ANOVA hypothesis test shows no significant differences. Instead, runs that consider only 5 features achieve nDCG@10 values that are significantly different (lower) with respect to the others. This is reasonable since by considering too few features, then there is a high information loss.

Figure 5 shows the nDCG@10 values and Annealing timings of the runs that used QA and SA. From this figure we can see that, in terms of efficiency (i.e., Annealing time), runs using QA required a shorter amount of time with respect to SA. On average, QA required ≈ 9.89 times less compared to SA, thus representing a more efficient alternative. Considering effectiveness, SA seems to be performing more consistently. However, on average it performs only ≈ 1.03 times better compared to QA.

Figure 6 shows how many times each feature has been kept by the participants' approaches using both QA and SA. In general, we can see that SA has been more selective leading to more consistent results. On the other hand, we can see that both approaches have kept the same features most of the time, indicating that these probably were the most informative features.
Teams adopted different approaches to address this task:

- team **BIT.UA** [20] tried different QUBO formulations that involved the usage of different correlation-based measures such as Spearman coefficient, Pearson coefficient, and Mutual Information [9]. Furthermore, their approach also involved the usage of a scaling factor to automatically balance the importance of the diagonal terms in the matrix Q with respect to the off-diagonal terms. Additionally, they also tried investigating some non-linear functions that adjusted the weights of the values returned by the correlation-based measures. The number of features chosen was decided by using a validation dataset approach with a custom LambdaMART model.

- team **NICA** [21] and team **shm2024** [22] used a QUBO formulation which involved the Mutual Information [9] as a correlation-based measure.

- team **QTB** [23] investigated different QUBO formulations involving different correlation-based measures (e.g., Mutual Information [9]). The team employed all methods (i.e., QA, H and SA), and the H approach allowed them to achieve a high score with only a few features thanks to its pre-processing and post-processing capabilities.

- team **OWS** [24] employed a QUBO matrix that was formulated using Mutual Information [9], in which some of its components were recalculated using the results achieved by a bootstrapping approach. In this way, the team recalculated the values associated with the diagonal components, the off-diagonal components, or both. The team focused on choosing only 25 features and the optimization of the number of considered features is left for future works.

5.1.2. Istella dataset.

As it is possible to see in Table 6 and in Figure 7, also in this case teams considered different numbers of features in their submissions. However, for the Istella dataset, most of the runs are statistically different from each other because the number of features used varies a lot. It is interesting to see that the baseline method employing Recursive Feature Elimination considering 110 features performed much worse with respect to all participants' runs. Furthermore, running Recursive Feature Elimination to keep the top

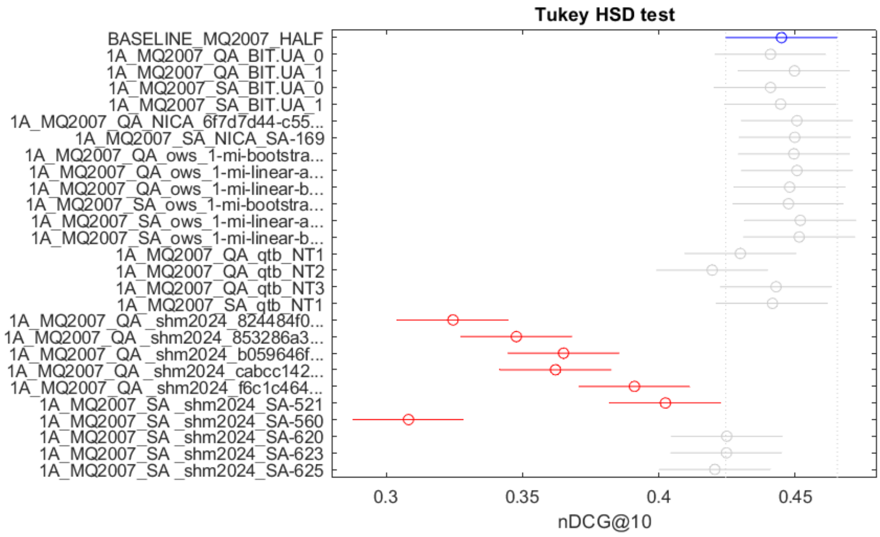
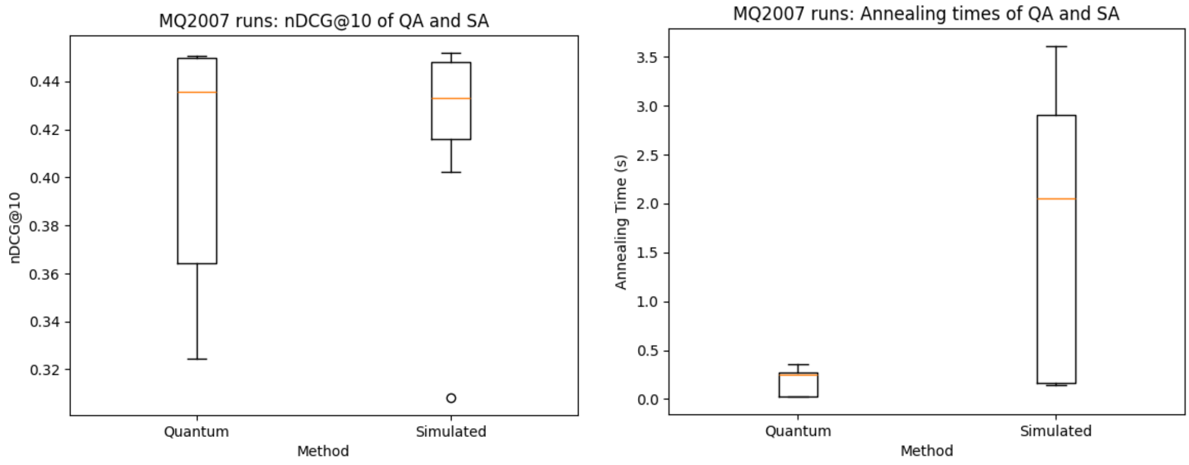


Figure 4: The Tukey HSD test considering the nDCG@10 values associated with different runs and queries for the MQ2007 dataset.



(a) nDCG@10 of SA and QA runs

(b) Annealing time of SA and QA runs

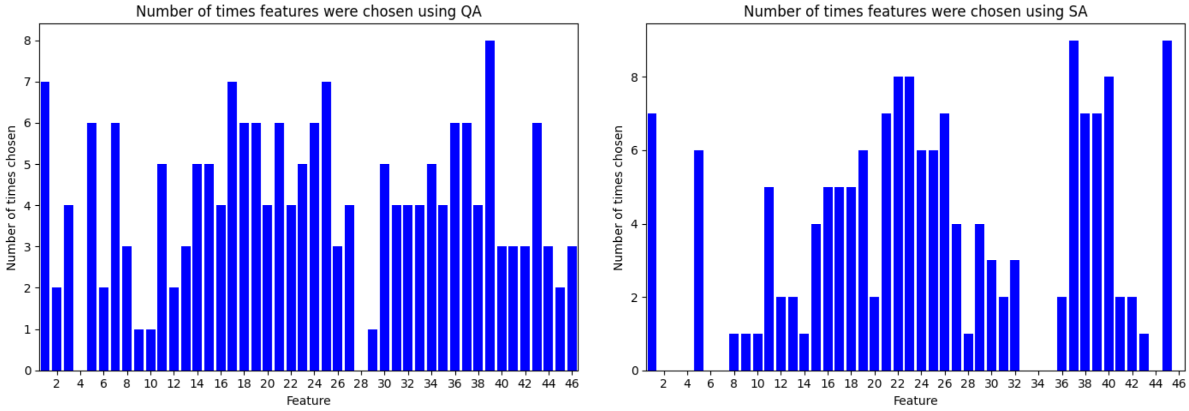
Figure 5: The box plots of the nDCG@10 values and Annealing timings associated with the runs using QA and SA on the MQ2007 dataset.

110 features required a considerable amount of time (almost 2 hours of computation) and a considerable amount of RAM (24 GB), which is much higher than the teams' workspace specifications.

The teams adopted similar approaches to the ones described for the MQ2007 dataset to solve the Feature Selection task on the Istella dataset. However, since the dataset could not fit entirely in the QPU due to the high number of features, two teams decided to adopt the following pre-processing techniques:

- team **BIT.UA** [20] employed different approaches such as using a first stage SA approach to select only a subset of features or the manual elimination of features with high correlation values between them before solving the problem with QA.
- team **NICA** [21] kept only the 50 features that had the highest Mutual Information value towards the target variable, thus reducing the feature set.

Figure 8 shows the nDCG@10 values and Annealing timings of the runs that used QA and SA. From this figure we can see that, in terms of efficiency (i.e., Annealing time), also in this case runs using



(a) Features kept using QA

(b) Features kept using SA

Figure 6: The number of times each feature of the MQ2007 dataset has been kept considering the different teams' approaches using QA and SA.

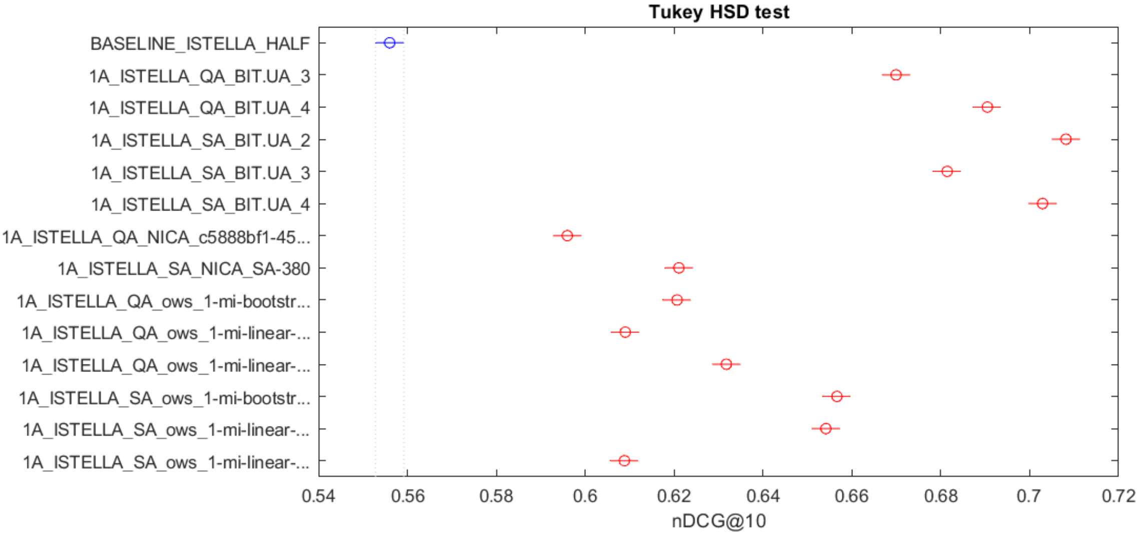


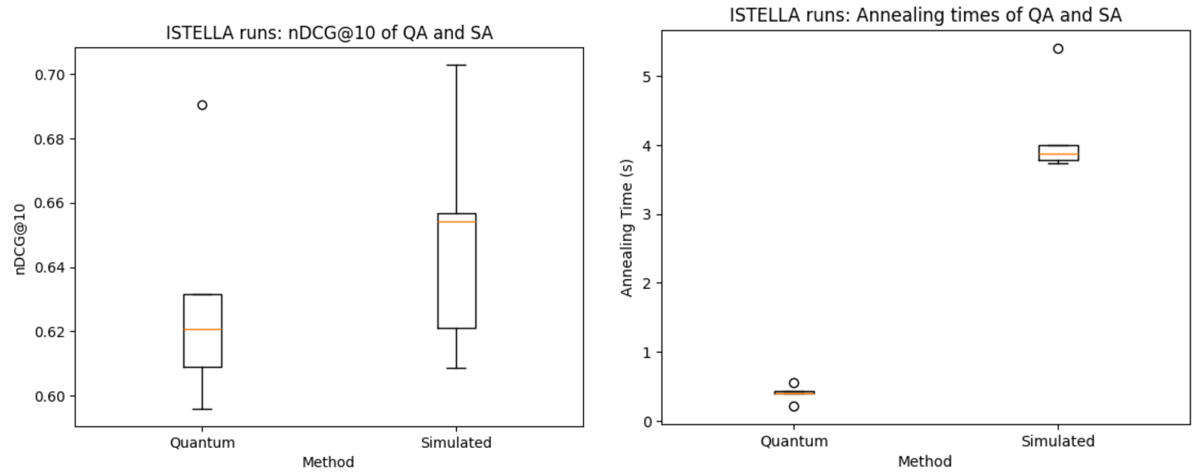
Figure 7: The Tukey HSD test considering the nDCG@10 values associated with different runs and queries for the Istella dataset.

QA required a shorter amount of time with respect to SA. On average, QA required ≈ 10.45 times less compared to SA, thus representing a more efficient alternative. Similar considerations apply also for effectiveness. In fact, SA seems to be performing more consistently however, on average it performs only ≈ 1.03 times better compared to QA.

Figure 9 shows how many times features have been kept by the participants' approaches using both QA and SA. For simplicity, we avoid showing the results for each feature since this would be very complicated to plot and read considering the 220 features of the Istella dataset. Overall, we can see similar trends as discussed for the MQ2007 dataset in which SA seems to be more selective but there is a shared set of features that were selected many times by both QA and SA.

5.2. Task 1B

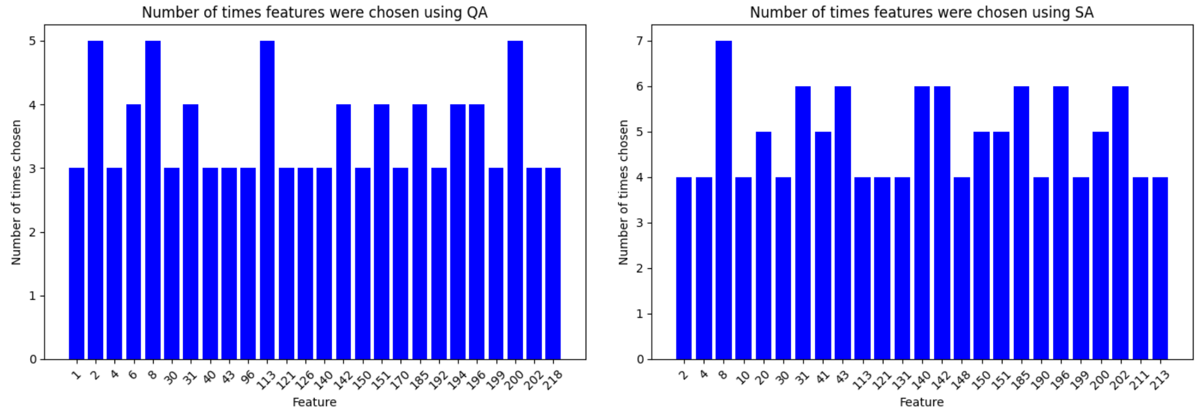
Here we present the results achieved by the two teams participating in task 1B. Results are divided according to the two feature sets. For both the small ICM (see Table 7) and the large one (see Table 8) the teams were able to improve the effectiveness of the baseline RS by a large margin, around 23% on the small set and 44% on the large one. Team **CRUISE** [25] especially achieved a large improvement by developing a counterfactual version of nDCG to enhance a feature selection method based on Mutual



(a) nDCG@10 of SA and QA runs

(b) Annealing time of SA and QA runs

Figure 8: The box plots of the nDCG@10 values and Annealing timings associated with the runs using QA and SA on the Istella dataset.



(a) Features kept using QA

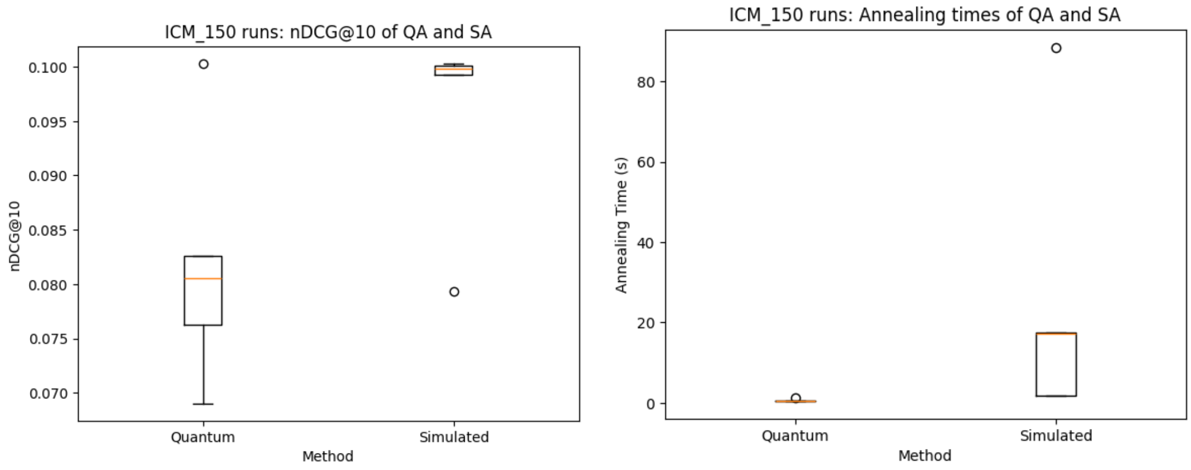
(b) Features kept using SA

Figure 9: The number of times each of the most kept features of the Istella dataset has been kept considering the different teams' approaches using QA and SA.

Information. The idea considers that Mutual Information does not account for the final goal of making recommendations.

The proposed approach is based on MIQUBO [9] and introduces a term in the diagonal of Q which represents the change in nDCG@10 obtained by removing each of the features individually, weighted by a scaling factor. In this way, the diagonal of Q includes both the Mutual Information between the feature values and the target label, as well as the weighted change in nDCG@10. For the small ICM, with 150 features, QA is 35.88 times faster than SA but it is 1.17 times worse in terms of nDCG@10 (see Figure 10).

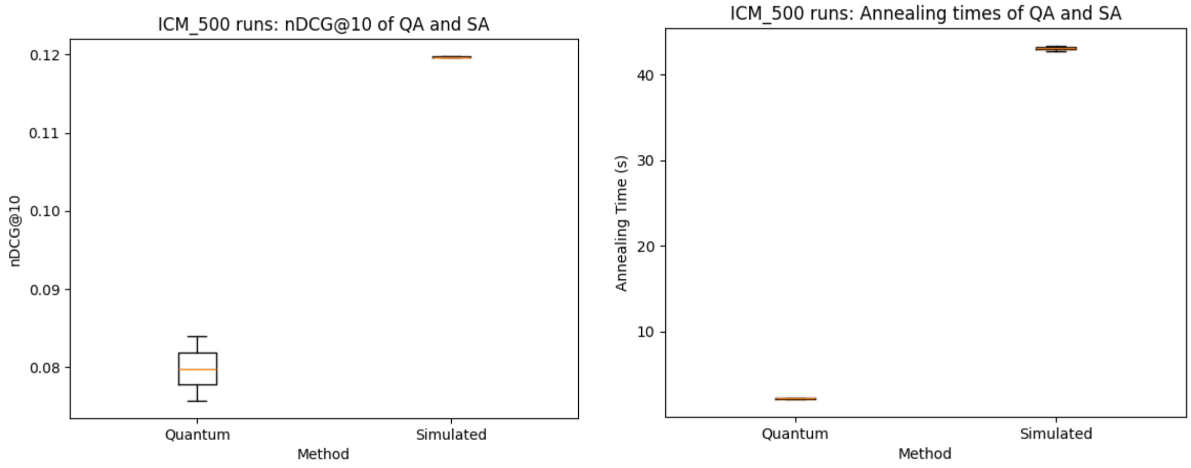
For the large ICM, with 500 features, that could not fit on the QPU, team **CRUISE** [25] split the features into subsets small enough to be tackled by the QPU. Then, the features selected in each subset have been merged into a final set of features. For this large ICM QA is 19.53 times faster than SA but it is 1.5 times worse in terms of nDCG@10 (see Figure 11). Note that the number of selected features is very different so this could play a role.



(a) nDCG@10 of QA and SA runs

(b) Annealing time of QA and SA runs

Figure 10: The box plots of the nDCG@10 values and Annealing timings associated with the runs using QA and SA on the ICM_150 dataset.



(a) nDCG@10 of QA and SA runs

(b) Annealing time of QA and SA runs

Figure 11: The box plots of the nDCG@10 values and Annealing timings associated with the runs using QA and SA on the ICM_500 dataset.

5.3. Task 2

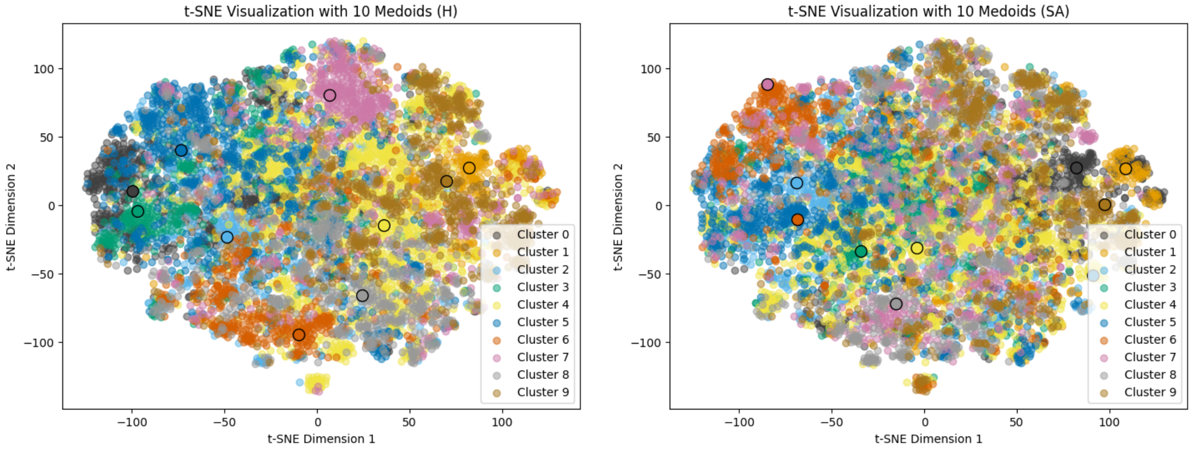
Here we present the results achieved by the teams participating in task 2. Table 9 reports the results achieved in this task.

In this task, we can see that team **qIIMAS** managed to achieve higher results with respect to the baseline for each number of clusters considered. The approach adopted by team **qIIMAS** [26] consisted of employing the QUBO formulation proposed in a previous work [27]. Due to the high dimensionality of the dataset, they decided to first apply a traditional approach to reduce the number of points n to some representatives m where $m < n$. Then they performed the clustering approach on the m representatives in a hierarchical fashion, returning the final set of centroids and their associated n points. They investigated the usage of both QA, H, and SA.

Figure 12 shows an example of the medoids and clusters found by the team **qIIMAS** for the runs submitted considering 10 clusters. Despite the high information loss associated with representing data in 2 dimensions using t-SNE, we can still observe that the Hybrid approach appears to produce qualitatively better clusters.

In Figure 13 we can observe that there are no statistical differences among runs using H and runs using SA considering the nDCG@10 values achieved, showing that H is indeed a robust approach that can be as effective as the SA counterpart.

Figure 14 shows the Annealing time of the runs that used H and SA. From this figure we can see that, in terms of efficiency (i.e., Annealing time), runs using H required a shorter amount of time with respect to SA. On average, H required ≈ 21.75 times less compared to SA, thus representing a more efficient alternative. In addition, the H methods achieved slightly better results in terms of effectiveness, being ≈ 1.02 times better than SA on average.



(a) t-SNE with 10 clusters using H

(b) t-SNE with 10 clusters using SA

Figure 12: The t-SNE visualizations of the 10 medoids and clusters of documents found in the submissions. Medoids are colored with the same colors as the corresponding clusters of documents but their edges are black.

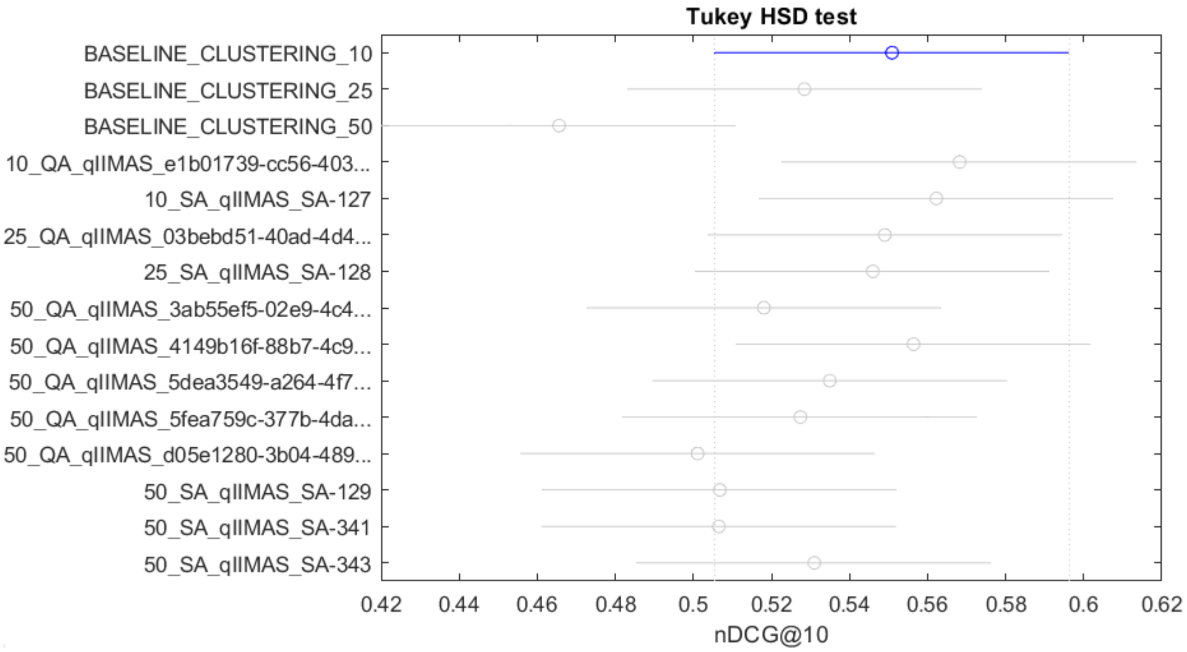
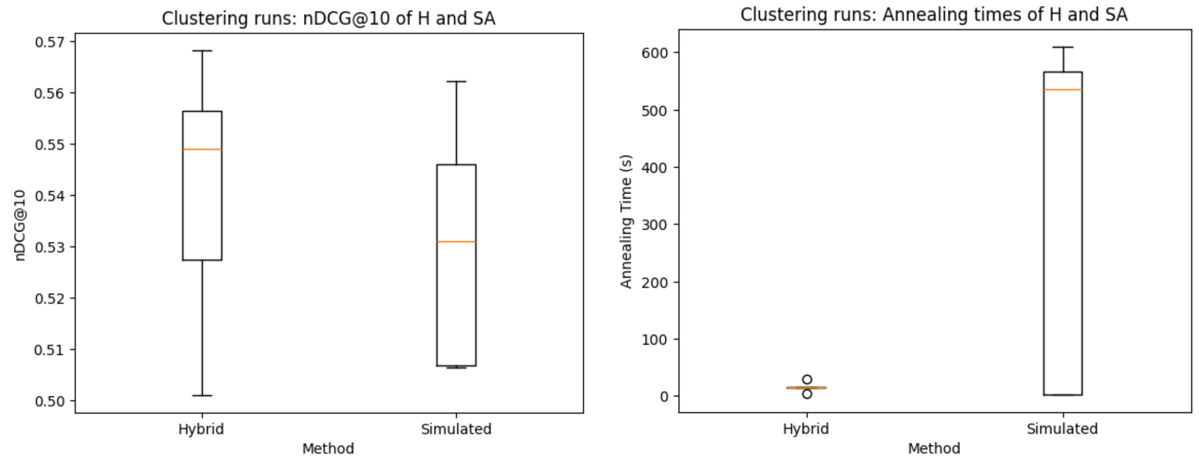


Figure 13: The Tukey HSD test considering the nDCG@10 values associated with different runs and queries for the Clustering dataset.



(a) nDCG@10 of H and SA runs

(b) Annealing time of H and SA runs

Figure 14: The box plots of the nDCG@10 values and Annealing timings associated with the runs using H and SA on the Clustering dataset.

6. Conclusions and Future Work

In this paper we have presented a comprehensive overview of the first edition of the QuantumCLEF 2024 lab, the first lab at CLEF dedicated to the exploration, development, and evaluation of QC algorithms.

This lab was composed of two tasks concerning the problems of Feature Selection and Clustering, specifically addressing challenges faced by IR and RS systems. An ad-hoc infrastructure was created to facilitate and streamline the participants' workflow and to grant them access to computational resources and the state-of-the-art quantum annealers provided by D-Wave.

A total of 26 teams registered for the lab and 7 of them successfully managed to submit their runs. The results have shown that QA and H managed to achieve comparable results in terms of effectiveness with respect to SA while achieving a higher level of efficiency in terms of Annealing time. These findings underscore the growing potential of QC, which can be now applied to solve practical problems. In fact, as QC evolves, we expect it to become more robust and powerful. In this way, it will be possible to employ it to find innovative solutions to a wider variety of computationally complex and real-world problems.

This lab represented a valuable opportunity not only to develop and evaluate QC algorithms on **real quantum computers** (quantum technologies are still not easily accessible to the general public) but also to raise awareness of the potential of QC, a technology that is likely to become very influential in the future. The data obtained throughout the challenge has also been fundamental in preparing a new QC tutorial presented to the international research community during the SIGIR conference 2024 [28].

Participants were supported with comprehensive materials such as videos, slides, and examples that allowed them to learn how QC and QA work. This ensured that participants could effectively engage with these innovative technologies.

Finally, we prioritized maximum transparency, allowing participants to work with the actual D-Wave libraries without constraining them to use custom functions. This hands-on experience with the official D-Wave tools enabled participants to become proficient in programming quantum annealers. As a result, they are now able to apply QC technologies beyond our lab environment to solve diverse problems within their own research fields.

In the future, we plan to organize a second edition of QuantumCLEF with different tasks and more challenging problems. We also plan to further improve the infrastructure according to the comments received by the participants through the lab to ensure a smoother experience for participants of a possible future edition of QuantumCLEF. Moreover, we would like to invest in a more powerful infrastructure that will grant access to more participants and that will provide more resources (in terms of CPU and RAM) to each workspace. In this way, it will be possible to consider even a more fair comparison between SA and QA. Furthermore, we aim to extend our infrastructure to incorporate gate-based quantum computers [29], alongside the currently available quantum annealers. By integrating gate-based quantum computers we could expand the family of problems that can be addressed within our QuantumCLEF lab, thus providing participants with a more diverse and powerful set of tools to explore and innovate in the realm of QC.

Acknowledgments

We acknowledge the financial support from ICSC - “National Research Centre in High Performance Computing, Big Data and Quantum Computing”, funded by the European Union – NextGenerationEU.

We acknowledge the CINECA award under the ISCRA initiative, for the availability of high-performance computing resources and support.

References

[1] A. Pasin, M. Ferrari Dacrema, P. Cremonesi, N. Ferro, qCLEF: A Proposal to Evaluate Quantum Annealing for Information Retrieval and Recommender Systems, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2023, pp. 97–108.

[2] A. Pasin, M. Ferrari Dacrema, P. Cremonesi, N. Ferro, QuantumCLEF-Quantum Computing at CLEF, in: European Conference on Information Retrieval, Springer, 2024, pp. 482–489.

[3] F. Glover, G. Kochenberger, R. Hennig, Y. Du, Quantum bridge analytics I: a tutorial on formulating and using QUBO models, Annals of Operations Research 314 (2022) 141–183.

[4] J. Cai, W. G. Macready, A. Roy, A practical heuristic for finding graph minors, arXiv preprint arXiv:1406.2741 (2014).

[5] S. Yarkoni, E. Raponi, T. Bäck, S. Schmitt, Quantum annealing for industry applications: Introduction and review, Reports on Progress in Physics (2022).

[6] D. Bertsimas, J. Tsitsiklis, Simulated annealing, Statistical science 8 (1993) 10–15.

[7] P. J. Van Laarhoven, E. H. Aarts, P. J. van Laarhoven, E. H. Aarts, Simulated annealing, Springer, 1987.

[8] M. Ferrari Dacrema, A. Pasin, P. Cremonesi, N. Ferro, Quantum Computing for Information Retrieval and Recommender Systems, in: European Conference on Information Retrieval, Springer, 2024, pp. 358–362.

[9] M. Ferrari Dacrema, F. Moroni, R. Nembrini, N. Ferro, G. Faggioli, P. Cremonesi, Towards Feature Selection for Ranking and Classification Exploiting Quantum Annealers, in: Proc. 45th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022), ACM Press, New York, USA, 2022, pp. 2814–2824.

[10] R. Nembrini, M. Ferrari Dacrema, P. Cremonesi, Feature selection for recommender systems with quantum computing, Entropy 23 (2021) 970.

[11] C. J. C. Burges, From RankNet to LambdaRank to LambdaMART: An Overview, Technical Report, Microsoft Research, MSR-TR-2010-82, 2010.

[12] T. Qin, T.-Y. Liu, Introducing letor 4.0 datasets, arXiv preprint arXiv:1306.2597 (2013).

[13] C. Lucchese, F. M. Nardini, S. Orlando, R. Perego, F. Silvestri, S. Trani, Post-learning optimization of tree ensembles for efficient ranking, in: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, 2016, pp. 949–952.

[14] Y. Wu, Q. Cao, H. Shen, S. Tao, X. Cheng, INMO: A model-agnostic and scalable module for inductive collaborative filtering, in: SIGIR ’22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, ACM, 2022, pp. 91–101. doi:10.1145/3477495.3532000.

[15] H. Ushijima-Mwesigwa, C. F. A. Negre, S. M. Mniszewski, Graph partitioning using quantum annealing on the d-wave system, CoRR abs/1705.03082 (2017). URL: http://arxiv.org/abs/1705.03082. arXiv:1705.03082.

[16] C. Bauckhage, N. Piatkowski, R. Sifa, D. Hecker, S. Wrobel, A QUBO formulation of the k-medoids problem, in: "Lernen, Wissen, Daten, Analysen", Berlin, Germany, volume 2454 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 54–63. URL: https://ceur-ws.org/Vol-2454/paper_39.pdf.

[17] D. Arthur, P. Date, Balanced k-means clustering on an adiabatic quantum computer, Quantum Inf. Process. 20 (2021) 294. doi:10.1007/s11128-021-03240-8.

[18] H. Hashemi, M. Aliannejadi, H. Zamani, W. B. Croft, Antique: A non-factoid question answering benchmark, in: Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42, Springer, 2020, pp. 166–173.

[19] L. Van der Maaten, G. Hinton, Visualizing data using t-sne., Journal of machine learning research 9 (2008).

[20] T. Almeida, S. Matos, Towards a hyperparameter-free qubo formulation for feature selection in ir, in: G. Faggioli, N. Ferro, P. Galuščáková, A. García Seco de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, 2024.

[21] A. Naebzadeh, S. Eetemadi, Nica at quantum computing CLEF tasks 2024, in: G. Faggioli, N. Ferro, P. Galuščáková, A. García Seco de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, 2024.

[22] S. Gersome, J. Mahibha, D. Thenmozhi, Team shm2024 on quantum feature selection, in: G. Faggioli, N. Ferro, P. Galuščáková, A. García Seco de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, 2024.

[23] E. Payares, E. Puertas, J. C. Martinez Santos, Team qtb on feature selection via quantum annealing and hybrid models, in: G. Faggioli, N. Ferro, P. Galuščáková, A. García Seco de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, 2024.

[24] M. Fröbe, D. Alexander, G. Hendriksen, F. Schlatt, M. Hagen, M. Potthast, Team openwebsearch at CLEF 2024: QuantumCLEF, in: G. Faggioli, N. Ferro, P. Galuščáková, A. García Seco de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, 2024.

[25] J. Niu, J. Li, K. Deng, Y. Ren, Cruise on quantum computing for feature selection in recommender systems, in: G. Faggioli, N. Ferro, P. Galuščáková, A. García Seco de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, 2024.

[26] W. Alvarez Giron, J. Tellez, J. Tovar Cortes, H. Gómez Adorno, Team qiimas on task 2 - clustering, in: G. Faggioli, N. Ferro, P. Galuščáková, A. García Seco de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, 2024.

[27] C. Bauckhage, N. Piatkowski, R. Sifa, D. Hecker, S. Wrobel, A qubo formulation of the k-medoids problem., in: LWDA, 2019, pp. 54–63.

[28] M. Ferrari Dacrema, A. Pasin, P. Cremonesi, N. Ferro, Using and Evaluating Quantum Computing for Information Retrieval and Recommender Systems (2024).

[29] E. Rieffel, W. Polak, An introduction to quantum computing for non-physicists, ACM Computing Surveys (CSUR) 32 (2000) 300–335.

A. Task 1A - Team Results

Table 5
The results for Task 1A on the MQ2007 dataset. An adjacent couple of rows (marked with the same color) represents the results achieved with QA/H and SA using the same problem formulation. Results marked in yellow(🟡) refer to the baselines' results.

Team	Submission id	nDCG@10	Annealing time (ms)	Type	N° features
BIT.UA	1A_MQ2007_QA_BIT.UA_0	0.441	274	QA	18
BIT.UA	1A_MQ2007_SA_BIT.UA_0	0.441	1351	SA	16
BIT.UA	1A_MQ2007_QA_BIT.UA_1	0.4497	270	QA	20
BIT.UA	1A_MQ2007_SA_BIT.UA_1	0.4446	3607	SA	18
NICA	1A_MQ2007_QA_NICA_6f7d7d44-c559-4e36-9b10-b7e51e521036	0.4506	274	QA	17
NICA	1A_MQ2007_SA_NICA_SA-169	0.4498	3510	SA	15
OWS	1A_MQ2007_QA_ows_1-mi-bootstrap-mixture	0.4495	279	QA	25
OWS	1A_MQ2007_SA_ows_1-mi-bootstrap-mixture	0.4475	2818	SA	25
OWS	1A_MQ2007_QA_ows_1-mi-linear-and-quadratic-bootstrapped-boost-3	0.4506	270	QA	25
OWS	1A_MQ2007_SA_ows_1-mi-linear-and-quadratic-bootstrapped-boost-3	0.4519	2752	SA	25
OWS	1A_MQ2007_QA_ows_1-mi-linear-bootstrapped-boost-3	0.448	241	QA	25
OWS	1A_MQ2007_SA_ows_1-mi-linear-bootstrapped-boost-3	0.4515	2759	SA	25
QTB	1A_MQ2007_QA_qtb_NT1	0.4299	356	QA	13
QTB	1A_MQ2007_SA_qtb_NT1	0.4024	3174	SA	10
QTB	1A_MQ2007_QA_qtb_NT2	0.4195	5000	H	10
QTB	-	-	-	SA	-
QTB	1A_MQ2007_QA_qtb_NT3	0.443	4309	H	10
QTB	-	-	-	SA	-
shm2024	1A_MQ2007_QA _shm2024_b059646f-a9fd-4fd6-9589-c6e117400a9e	0.365	30	QA	5
shm2024	1A_MQ2007_SA _shm2024_SA-521	0.4024	284	SA	5
shm2024	1A_MQ2007_QA _shm2024_cabcc142-3fc5-4b22-8a6b-c7a45857fbc2	0.3621	27	QA	5
shm2024	1A_MQ2007_SA _shm2024_SA-560	0.3082	164	SA	5
shm2024	1A_MQ2007_QA _shm2024_f6c1c464-6dba-4a44-93b8-92ad6c4f60f9	0.391	29	QA	5
shm2024	1A_MQ2007_SA _shm2024_SA-620	0.4249	143	SA	5
shm2024	1A_MQ2007_QA _shm2024_853286a3-7f47-4de8-b0a0-247a65e6f6b6	0.3477	28	QA	5
shm2024	1A_MQ2007_SA _shm2024_SA-623	0.4248	147	SA	5
shm2024	1A_MQ2007_QA _shm2024_824484f0-b6fa-44b6-9bc7-0cb073db84e7	0.3245	29	QA	5
shm2024	1A_MQ2007_SA _shm2024_SA-625	0.4205	144	SA	5
BASELINE	ALL_FEATURES	0.4473	-	-	46
BASELINE	RFE HALF_FEATURES	0.4450	-	-	23

Table 6
The results for Task 1A on the Istella dataset. An adjacent couple of rows (marked with the same color) represents the results achieved with QA/H and SA using the same problem formulation. Results marked in yellow(🟡) refer to the baselines' results.

Team	Submission id	nDCG@10	Annealing time (ms)	Type	N° features
BIT.UA	1A_Istella_QA_BIT.UA_3	0.6699	16325	SA+QA	92
BIT.UA	1A_Istella_SA_BIT.UA_3	0.6814	19071	SA	90
BIT.UA	1A_Istella_QA_BIT.UA_4	0.6905	551	QA	82
BIT.UA	1A_Istella_SA_BIT.UA_4	0.7029	5404	SA	72
BIT.UA	-	-	-	SA	-
BIT.UA	1A_Istella_SA_BIT.UA_2	0.7081	13827	SA	161
NICA	1A_Istella_QA_NICA_c5888bf1-4549-418c-92b8-b7175c9185e4	0.596	427	QA	15
NICA	1A_Istella_SA_NICA_SA-380	0.6211	3998	SA	15
OWS	1A_Istella_QA_ows_1-mi-bootstrap-mixture	0.6207	215	QA	25
OWS	1A_Istella_SA_ows_1-mi-bootstrap-mixture	0.6566	3875	SA	25
OWS	1A_Istella_QA_ows_1-mi-linear-and-quadratic-bootstrapped-boost-3	0.609	394	QA	25
OWS	1A_Istella_SA_ows_1-mi-linear-and-quadratic-bootstrapped-boost-3	0.6541	3728	SA	25
OWS	1A_Istella_QA_ows_1-mi-linear-bootstrapped-boost-3	0.6317	402	QA	25
OWS	1A_Istella_SA_ows_1-mi-linear-bootstrapped-boost-3	0.6088	3785	SA	25
BASELINE	ALL_FEATURES	0.7146	-	-	220
BASELINE	RFE HALF_FEATURES	0.5560	-	-	110

B. Task 1B - Team Results

Table 7

Task 1B results on the 150_ICM dataset. Adjacent row pairs (same color) show the results achieved with QA/H and SA for the same problem formulation. Results highlighted in yellow(🟡) refer to the baselines' results.

Team	Submission id	nDCG@10	Annealing time (ms)	Type	N° features
CRUISE	1B_150_ICM_QA_CRUISE_1	0.0805	536	QA	138
CRUISE	1B_150_ICM_SA_CRUISE_1	0.0998	1745	SA	140
CRUISE	1B_150_ICM_QA_CRUISE_2	0.0826	529	QA	136
CRUISE	1B_150_ICM_SA_CRUISE_2	0.0993	17358	SA	140
CRUISE	1B_150_ICM_QA_CRUISE_3	0.0690	531	QA	132
CRUISE	1B_150_ICM_SA_CRUISE_3	0.1001	1760	SA	140
CRUISE	1B_150_ICM_QA_CRUISE_4	0.0763	558	QA	133
CRUISE	1B_150_ICM_SA_CRUISE_4	0.0793	17387	SA	140
CRUISE	1B_150_ICM_QA_CRUISE_5	0.1003	1375	QA	144
CRUISE	1B_150_ICM_SA_CRUISE_5	0.1003	88395	SA	144
NICA	-	-	-	QA	-
NICA	1B_150_ICM_SA_NICA_SA-457	0.0895	12247	SA	145
BASELINE	ALL_FEATURES	0.0810	-	-	150

Table 8

Task 1B results on the 500_ICM dataset. Adjacent row pairs (same color) show the results achieved with QA/H and SA for the same problem formulation. Results highlighted in yellow(🟡) refer to the baselines' results.

Team	Submission id	nDCG@10	Annealing time (ms)	Type	N° features
CRUISE	1B_500_ICM_QA_CRUISE_1	0.0757	2287	QA	407
CRUISE	1B_500_ICM_SA_CRUISE_1	0.1196	43339	SA	450
CRUISE	1B_500_ICM_QA_CRUISE_2	0.0839	2123	QA	397
CRUISE	1B_500_ICM_SA_CRUISE_2	0.1198	42777	SA	450
BASELINE	ALL_FEATURES	0.0827	-	-	500

C. Task 2 - Team Results

Table 9

Task 2 results. Adjacent row pairs (same color) show the results achieved with QA/H and SA for the same problem formulation. Results highlighted in yellow(🟡) refer to the baselines' results.

				Annealing		
Team	Submission id	nDCG@10	DBI	time (ms)	Type	N° centroids
qIIMAS	10_QA_qIIMAS_e1b01739-cc56-4034-baa2-558a44e0bd65	0.5682	6.3121	14993	H	10
qIIMAS	10_SA_qIIMAS_SA-127	0.5622	6.6847	535516	SA	10
BASELINE	BASELINE_10	0.5509	7.9892	-	-	10
qIIMAS	25_QA_qIIMAS_03bebd51-40ad-4d45-b2a1-8d58b829afc6	0.549	5.3510	14995	H	25
qIIMAS	25_SA_qIIMAS_SA-128	0.546	5.3369	565985	SA	25
BASELINE	BASELINE_25	0.5284	6.1201	-	-	25
qIIMAS	50_QA_qIIMAS_4149b16f-88b7-4c90-b9cd-7bec94929898	0.5564	4.8032	14993	H	50
qIIMAS	50_SA_qIIMAS_SA-129	0.5068	4.7868	610068	SA	50
qIIMAS	50_QA_qIIMAS_5fea759c-377b-4dab-80d8-ca70ba205f02	0.5274	4.9537	29995	H	50
qIIMAS	50_SA_qIIMAS_SA-343	0.531	4.8112	2871	SA	50
qIIMAS	50_QA_qIIMAS_d05e1280-3b04-4897-aa0f-5d508d72d8e0	0.5011	5.0868	3994	H	50
qIIMAS	50_SA_qIIMAS_SA-341	0.5065	5.4160	2793	SA	50
qIIMAS	50_QA_qIIMAS_3ab55ef5-02e9-4c4d-9c7c-51cb56be3d9a	0.518	5.1842	18	QA	50
qIIMAS	-	-	-	-	SA	50
qIIMAS	50_QA_qIIMAS_5dea3549-a264-4f70-812a-ac52b2108663	0.5349	4.6978	67	QA	50
qIIMAS	-	-	-	-	SA	50
BASELINE	BASELINE_50	0.4656	5.3679	-	-	50