

UNIVERSITÀ
DEGLI STUDI
DI PADOVA



DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

MASTER THESIS IN COMPUTER ENGINEERING

QuantumCLEF: A Shared-Task Proposal to Evaluate the Performance of Quantum Computing for Information Retrieval Systems

MASTER CANDIDATE

Andrea Pasin

Student ID 2041605

SUPERVISOR

Prof. Nicola Ferro

University of Padua

ACADEMIC YEAR
2022/2023

Voglio qui ringraziare diverse persone che hanno avuto un ruolo di fondamentale importanza nella mia vita e che mi hanno permesso di fare le scelte che mi hanno portato ad essere qui oggi.

Innanzitutto desidero ringraziare con tutto il cuore la mia famiglia per essermi stata di supporto anche durante i momenti difficili.

Ringrazio immensamente anche Anna, che mi ha accompagnato durante questo percorso universitario sostenendomi sempre e spingendomi a dare il meglio di me.

Ovviamente un grazie speciale va anche ai miei amici che hanno alleggerito questo percorso universitario che a volte è stato assai tortuoso.

Un ulteriore grazie va al professor Giuseppe Tomiello che durante la scuola superiore è stato per me di fondamentale importanza poichè mi ha trasmesso la passione e l'entusiasmo verso l'informatica spronandomi a proseguire con il percorso di studi universitario.

Voglio infine ringraziare il professor Nicola Ferro per la sua disponibilità, competenza, gentilezza e per i suoi preziosi consigli.

Abstract

Quantum Computing has been a focus of research for many researchers over the last few years. As a result of technological development, nowadays Quantum Computing resources are becoming available and usable to solve practical problems also in the Information Retrieval (IR) field.

In this work, we firstly dive into the paradigms of Universal Quantum Computing and, in particular, Quantum Annealing which is the main focus. We also show how problems such as Feature Selection, a well-known *NP*-Hard problem, can be formulated as Quadratic Unconstrained Binary Optimization (QUBO) problems and embedded into Quantum Annealers.

Then we propose some possible Shared Tasks to evaluate the efficiency and effectiveness of Quantum Computing in the Information Retrieval field. These tasks will be proposed in the future to CLEF in order to start the QuantumCLEF evaluation campaign whose aim is to acknowledge the potential benefits of Quantum Annealing technologies in the IR field and to create a common ground for the research community to start learning and employing these precious resources to improve the current state-of-the-art solutions.

Finally we design and implement a Submission System that can be employed in order to carry out the Shared Tasks. This system is designed to be scalable, secure and fault-tolerant.

Sommario

Il Quantum Computing, o calcolo quantistico, è stato un settore verso il quale molti ricercatori hanno incentrato il loro interesse negli ultimi anni. Grazie allo sviluppo tecnologico, al giorno d'oggi le risorse di calcolo quantistico stanno diventando disponibili e fruibili per risolvere problemi pratici anche nel campo dell'Information Retrieval (IR).

In questo progetto innanzitutto esploriamo i paradigmi dell'Universal Quantum Computing ed, in particolare, del Quantum Annealing che sarà l'obiettivo principale. Inoltre mostriamo come alcuni problemi come quello di Feature Selection, un noto problema di tipo *NP-hard*, possono essere formulati come problemi QUBO, ossia problemi di ottimizzazione binaria quadratica non vincolata, ed infine risolti mediante dispositivi di Quantum Annealing.

Dopodichè, proponiamo alcune Shared Tasks per valutare l'efficienza e l'efficacia del Quantum Computing nel settore dell'Information Retrieval. Queste sorte di sfide verranno proposte in futuro a CLEF per dare inizio alla campagna di valutazione denominata QuantumCLEF. Questa avrà come scopo quello di capire quali sono i benefici che possono essere tratti dalle tecnologie di Quantum Annealing ed inoltre di creare una base comune per la comunità di ricerca per iniziare ad apprendere e impiegare queste risorse preziose allo scopo di migliorare il corrente stato dell'arte.

Infine progettiamo e implementiamo un sistema di sottomissione che può essere impiegato per svolgere queste Shared Tasks. Questo sistema dovrà essere scalabile, sicuro e a prova di errore.

Contents

List of Figures	xi
List of Tables	xiii
List of Algorithms	xvii
List of Code Snippets	xvii
List of Acronyms	xix
1 Introduction	1
2 Related Work	5
2.1 Information Retrieval	5
2.1.1 History of IR	5
2.1.2 Evaluation of IR systems	9
2.1.3 Applications of IR systems	13
2.2 Quantum Computing	15
2.2.1 Introduction to Computational Complexity	15
2.2.2 Main Concepts of Quantum Computing	19
2.2.3 Quantum Annealing	20
2.2.4 Quantum Annealing: Environmental Considerations	27
2.2.5 Applications of Quantum Computing	28
2.3 Evaluation Campaign	30
2.3.1 Shared Tasks and their Importance	30
2.3.2 CLEF	31
3 Shared Tasks: Overview	33
3.1 QUBO formulation	33

CONTENTS

3.2	Feature Selection	38
3.3	Other Quantum Annealing problems	44
3.3.1	Clustering	44
3.3.2	Boosting	46
4	Shared-Task Proposals	51
4.1	Task 1: Feature Selection	51
4.2	Task 2: Clustering	52
4.3	Task 3: Boosting	60
4.4	Additional Sub-Tasks	60
4.5	Uncertainties of Quantum Annealing	61
4.6	Efficiency of Quantum Annealing	61
5	Design of the Infrastructure	65
5.1	The Submission System	65
5.2	Our Submission System as a Distributed System	68
5.3	Container orchestration: Kubernetes	70
5.4	Web Application	72
5.4.1	Web Application: Database	72
5.4.2	Web Application: Back End and Front End	73
5.4.3	Web Application: security issues	75
6	Implementation of the System	79
6.1	Web App Implementation	79
6.2	The Submission System from a Kubernetes Point of View	84
6.3	An in-depth View into each Container	88
6.4	The Submission System running	91
7	Conclusions and Future Works	95
	References	97
	Acknowledgments	103

List of Figures

2.1	A representation of Goldberg’s Statistical Machine.	6
2.2	The design of the Memex (Bush, 1945, p. 123).	7
2.3	Graphical representation where $f(n) = O(g(n))$	16
2.4	Graphical representation of the classes of problems.	18
2.5	Representation of thermal cooldown of an object with respect to its environment.	22
2.6	Representation of how a qubit’s state is implemented in a D-Wave quantum annealer.	23
2.7	A very simple representation of the Annealing process involving a single qubit.	24
2.8	Energy landscape affected by bias.	25
2.9	Representation of couplers.	26
2.10	An example of Quantum Annealing considering 2 qubits coupled together.	26
3.1	Portion of the Pegasus topology employed in the D-Wave Ad- vantage QPUs. This figure has been obtained through D-Wave NetworkX Python library [45].	37
3.2	Representation of the process of minor embedding.	38
3.3	Graphical representation of the Feature Selection process.	38
3.4	Example of underfitting and overfitting.	40
3.5	Representation of k-means with $k = 3$	46
3.6	Representation of a Random Forest.	47
3.7	Representation of a Decision Tree.	47
4.1	The graphical interface with the D-Wave inspector tool that shows how the problem has been embedded in the QPU.	53

LIST OF FIGURES

4.2	Example of the Iterative Clustering approach. This can be used to solve the issue of fitting the Clustering problem on the QPU that has a limited number of qubits and connections available.	59
4.3	Representation of the various Quantum Processing Unit (QPU) times corresponding to the different phases when solving a problem through the D-Wave quantum annealer.	63
5.1	Representation of the design of the submission system. Docker containers are being used in order to make everything more scalable and secure.	66
5.2	Representation of how the communication takes place in the Submission System starting from the personal machine of a group to the quantum annealer.	67
5.3	An example of the difference between the deployment of some applications on a host machine, on virtual machines and on containers.	69
5.4	Entity-Relationship schema of the SQL database used to manage the Web Application.	73
5.5	Logical schema of the SQL database used to manage the Web Application.	73
5.6	An example of an Hacker sniffing the communication between the User and the Web Application.	74
5.7	A simple example in which it is possible to see the communication between the User, Server and Database.	75
6.1	The homepage of our Website.	81
6.2	The form to apply for a task.	81
6.3	The login form to access the protected area.	82
6.4	A first view into the groups' protected area.	83
6.5	A second view into the groups' protected area.	83
6.6	A view into the administrator section where it is possible to have a look at the groups who have applied to the currently active tasks and modify the corresponding data.	84
6.7	The representation of our Submission System based on some of the most important Kubernetes objects.	84
6.8	The Kubernetes deployments of our Submission System.	92

6.9	The interface that each group has representing its corresponding workspace if using the Visual Studio Code IDE.	92
6.10	The group's private area where all the submissions are tracked by the Dispatcher.	93

List of Tables

2.1	Table representing the time required based on complexity.	16
3.1	Table representing some useful conversions between constraints to corresponding penalties.	36

List of Algorithms

1	Naive Feature Selection Algorithm.	41
2	Naive approach to Feature Selection improved.	42
3	Boosting applied to Random Forests.	48
4	Approximation of the Clustering problem.	58

List of Code Snippets

4.1	Code snippet related to the Clustering problem.	53
5.1	SQL statement to retrieve the number of users having the specified username and password.	76
5.2	SQL statement with SQL injection applied.	76
5.3	SQL statement with SQL injection applied.	76
5.4	Example of a Cross-Site Scripting attack.	77
6.1	The Kubernetes yaml code for the Deployment and Service of the PostgreSQL database.	86
6.2	The Dockerfile used to create the container image for the groups.	90
6.3	The command to build the Docker image for the group's workspace container.	91

List of Acronyms

IR Information Retrieval

QC Quantum Computing

QA Quantum Annealing

QUBO Quadratic Unconstrained Binary Optimization

BQM Binary Quadratic Model

TSP Travelling Salesman Problem

MI Mutual Information

QPU Quantum Processing Unit

IDE Integrated Development Environment

ER Entity-Relationship

DBMS Database Management System

ORM Object Relational Mapper

XSS Cross Site Scripting

DOM Document Object Model

CSRF Cross-Site Request Forgery

DS Distributed System

VM Virtual Machine

OS Operating System

ANOVA Analysis of Variance



Introduction

Everyone of us is making use as a daily basis of Search Engines to find any sort of information. Even when we just say "Hey Google, what's the weather like in Padua?" or "Alexa, can you tell me a joke?" we are basically employing systems that have been developed in order to answer to some information needs. There is a specific research field called Information Retrieval (Information Retrieval (IR)) that deals with the process of obtaining resources which are relevant to an information need. Usually the relevant resources are searched and retrieved through a collection of resources that can be arbitrarily large.

As you can imagine, this field is very important since it provides a way to find the answers that we need without having to spend a lot of time by potentially going through several books or articles before finding even very simple answers. To raise your awareness of the importance of this field we can think of the following situations.

Imagine being a librarian and a student asks you for a book about Leonardo Da Vinci. Then you need to go through the shelves that you think to be holding some possible books related to him. If you work in a very big library, it can take quite a lot of time to look through the shelves even if they had been organized in a clever way to help quickly finding the books you are looking for. In this case, an IR system that automatically returns you the title and the positions of the books containing data about Leonardo Da Vinci could be very useful to save up time and be more effective in finding what the student needs.

But now let's consider a different kind of situation. Imagine being a doctor that

needs to browse an archive of potential illnesses and corresponding medical treatments because a patient is not feeling well at all. Would you prefer having a system that helps you in finding the information you need to understand how to treat the patient almost immediately or would you rather go through collections of images or data by hand? Here time can be crucial to save a person's life!

In the aforementioned situations it is clear that time plays an important role in our lives. In fact, an IR system is valuable if it is able to provide answers in a small amount of time, namely it is **efficient**.

But is it enough? Of course no! Imagine an IR system that provides results in just a bunch of microseconds but the results themselves are not relevant with respect to what was asked to the system. In this case we cannot say that the system is performing well.

In fact, an Information Retrieval system needs to be both **efficient** and **effective** in order to satisfy the user needs.

Assessing whether a system is actually performing well or not is generally very difficult to be established. In fact, effectiveness can be subjective thus making evaluation very complex to be done.

To overcome this issue, experimental evaluation is usually conducted through large-scale evaluation campaigns in which several systems are submitted by many research groups and judges evaluate them according to some chosen metrics.

As you can imagine, IR systems can be quite complex to be implemented and usually they are required to work with very large amount of data. You can think of the Google Search Engine, which deals with billions of Web pages or maybe the Spotify's Music Recommendation systems that are serving millions of users every day by providing songs tailored to the users' preferences.

In this modern scenario where the amount of available data keeps growing and thus problems getting more and more complex, Quantum Computing technologies can represent a possible solution to implement more powerful IR systems. Quantum Computing is a branch of Computer Science and Quantum Physics that deals with studying and creating devices exploiting quantum mechanics phenomena. It has been shown that Quantum computers can perform some calculations exponentially faster with respect to any normal computer.

In the latest years Quantum Computing has become more practical and several Quantum Computing devices have been implemented. This has led to an increase in the availability of these devices for both research purposes and practical experiments.

The aim of this work is to explore the potential benefits of Quantum Computing and, in particular, Quantum Annealing applied to the Information Retrieval field. This is, in fact, an area that has not been researched much so far. Therefore we want to further investigate in order to understand how Quantum Annealing performs with respect to more classical approaches. In order to assess whether Quantum Annealing can be used to provide better solutions to the current problems, we propose here the QuantumCLEF evaluation campaign. As already mentioned, evaluation campaigns play a fundamental role in the IR field.

The campaign will consist in 3 different Shared Tasks which require participants to find some formulations of the given problems such that they can be solved by means of a quantum annealer.

Only by comparing the different solutions provided by the participants it will be possible to evaluate their efficiency and effectiveness with respect to the classical solutions that are employed nowadays.

In addition, it will be possible for researchers to start learning about Quantum Annealing with the actual employment of cutting edge quantum annealers.

In this work, we are firstly going to have an overview of what an Information Retrieval system is, how its performances can be evaluated according to some important metrics in the IR field and some practical applications of IR systems nowadays.

Furthermore we will introduce some Quantum Computing fundamental concepts and we will have a deeper insight regarding the Quantum Annealing paradigm and its possible applications.

In addition, we will also have a look at how an evaluation campaign usually takes place and we will introduce CLEF, the European large-scale evaluation campaign which involves more than 200 research groups from different nations. After having introduced the aforementioned topics and concepts, in Chapter 3 we will discuss about how to formulate problems in order to be solved by quantum annealers. We will see some practical examples of problems formulated according to their Quadratic Unconstrained Binary Optimization expression. In

addition, we will propose 3 problems that can be solved by quantum annealers: Feature Selection, Clustering and Boosting.

In Chapter 4 we will formalize a set of 3 possible Shared Tasks according to the problems investigated in Chapter 3. We will also make some considerations about effectiveness and efficiency of Quantum Annealing in these tasks.

Moreover, in Chapter 5 we will design a Submission System that can be used to handle the submissions of the participants to the tasks. We will provide an high level overview of the system and we will talk about potential security vulnerabilities.

In Chapter 6 we will provide an implementation of the Submission System using several different technologies, from Containers to specific tools used to develop the Database and Web Application.

Finally in Chapter 7 we will draw some conclusions and considerations about the work that has been done. We will also talk about future work regarding this evaluation campaign.

The final and most important objective of this work is to present a possible solution that will be implemented in the close future so that the QuantumCLEF evaluation campaign will be started in 2024. This campaign will hopefully be a starting point where research groups from different fields (e.g. Information Retrieval, Operations Research, Quantum Computing...) will find a common ground to design new solutions and compare their own solutions with the ones of other research groups. This will allow to assess which is the current state of the art and thus make improvements for a brighter future.



Related Work

In this chapter we are going to introduce you to the field of Information Retrieval by providing you an overview about its history and applications. Then we will have an overview about the Quantum Computing field, exploring the most important concepts with particular accents about the Quantum Annealing (QA) paradigm. Finally we will explain what is an evaluation campaign, how it usually takes place and why it is so important to have evaluation campaigns in the field of Information Retrieval.

2.1 INFORMATION RETRIEVAL

Information Retrieval is a research field that deals with the process of finding relevant resources corresponding to some information needs. IR systems are employed in several fields and they are developed in order to be as most efficient and effective as possible to automate and simplify the process of finding information resources.

2.1.1 HISTORY OF IR

Dealing with large collections of resources has been a problem for centuries. In fact, it is believed that the first conventional approaches of managing these collections originated from the discipline of librarianship. These approaches

2.1. INFORMATION RETRIEVAL

consisted in using some cataloging schemes. These schemes were employed to correctly index books and volumes in order to be able to retrieve them later with ease [1].

Mechanical approaches have been invented in the 20th century, such as the Statistical Machine invented by Emanuel Goldberg [2].

This machine was an electromechanical machine that could be employed for searching through data encoded on reels of film. In Goldberg's basic design a sort of "search card" is created and placed between a light source and the given film. The search card blocks all light from the light source except for a pattern of beams defining the code that is to be sought. Beyond the film there is a photocell. As the film containing images of documents moves through the machine, some of the light that passes through the search card will also pass through the film and finally reach the photocell, thus generating an electrical signal that can be measured by means of opportune circuitry. When opaque dots on the film coincide exactly with the pattern of light beams defined by the search card, all light is blocked and thus no light reaches the photocell. In that case, circuitry detects the loss of current and indicates to the user that the desired document has been found [3].

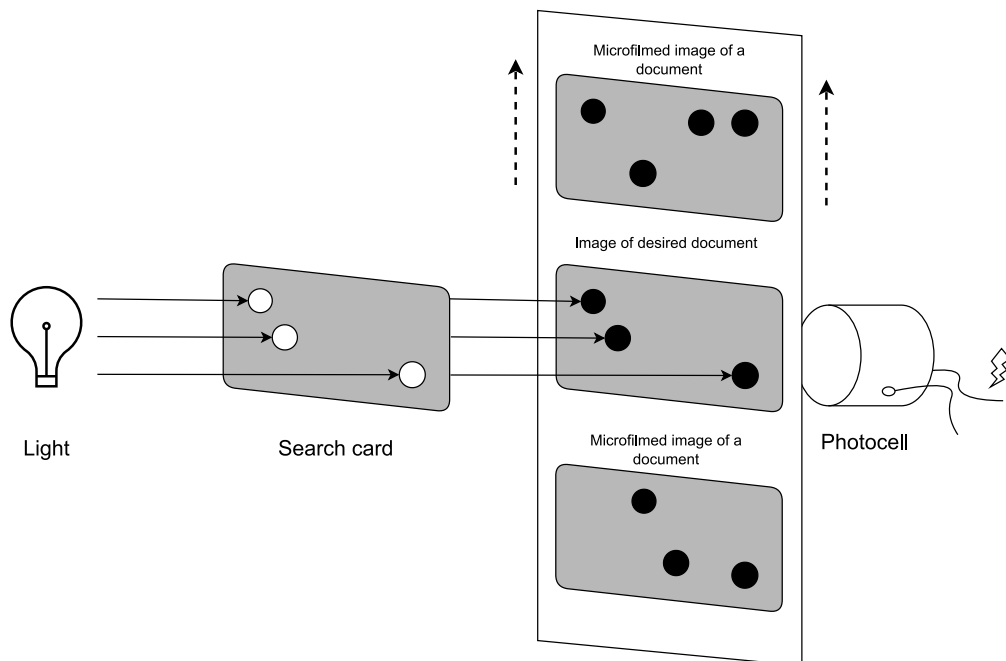


Figure 2.1: A representation of Goldberg's Statistical Machine.

In 1945 Vannevar Bush presented the idea of Memex [4], which was an electromechanical device that could be used to automatically store books, documents and any sort of personal record or annotation. This was taught to be a personal system that could be used to manage the memory of an individual. In fact, the name Memex is a portmanteau word of *Memory* and *Expansion*. The Memex idea was revolutionary for that epoch and was able to influence the development of the future Hypertext and Information Retrieval systems. The Memex was indeed the first idea of a complex Information Retrieval system that could be used both to keep a permanent storage of the human knowledge and to browse for it when needed in an automatic way.

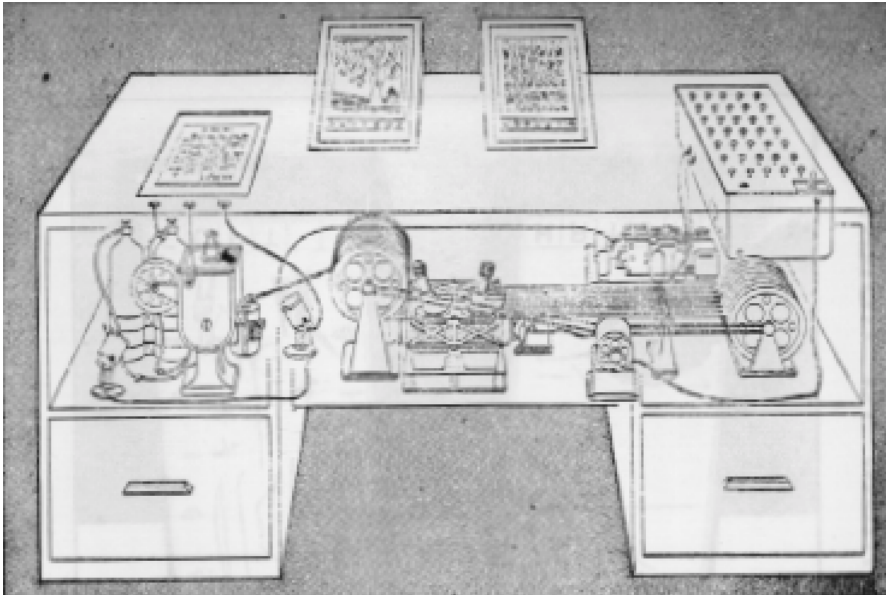


Figure 2.2: The design of the Memex (Bush, 1945, p. 123).

In the 1950s, scientists started to employ computers for IR approaches. In fact, they managed to implement solutions in order to look through collections of even thousands of records. At that time, it appeared clear that it was fundamental to understand how to index documents and retrieve them in the most efficient and effective way possible.

In 1952 a ground-breaking idea was proposed [5]. This idea was about indexing resources by means of a list of keywords rather than the classical hierarchical classification scheme that had been employed by librarians for centuries. This approach was proven to provide better results through experimental evaluation

2.1. INFORMATION RETRIEVAL

and it is still currently used nowadays in many IR systems.

In the same period, researchers also proposed a new way to retrieve documents that provided better results in terms of effectiveness: the Ranked Retrieval approach [6]. The classical Boolean Retrieval approach consisted in having queries that could be seen as Boolean combinations of terms. The systems then returned only the documents that were matching the given queries completely. The Ranked Retrieval approach instead was a probabilistic approach in which keywords related to each document were weighted based on their importance related to the associated document. This allowed to retrieve documents by assigning them scores and then ordering them consequently from the one having the highest score with respect to the given query to the one having the lowest score.

In the next couple of decades several innovations were brought to life such as:

- Relevance feedback [7]: technique that improves the process of iterative search (i.e. a user issuing sequential queries to fine tune the search scope) by adjusting the submitted queries by extracting information from the previously retrieved documents that had been perceived as relevant by the user.
- Stemming: technique in which multiple terms are associated to the same stem based on their meaning and spelling. Several stemmers have been created and the most famous ones include the Porter's stemmer [8] and the Lovins stemmer [9].
- Vector Space Model [10]: model in which each document can be represented as a vector which lies into an N -dimensional space where N is the number of distinct terms in the overall collection of documents. This allows to compute vector similarity measures between the submitted query and the documents (e.g. the Cosine Similarity) that will be used to score the documents according to how similar they are with respect to the given query.

Until the 1990s scientists were focused on fine tuning manually the weights of the proposed ranking functions (e.g. BM25 [11]) by adjusting them based on the experimental results. This was generally a very complex and time consuming process since it required to tune the parameters manually based on the considered collection [1]. It then became obvious that it was infeasible to apply this idea in the Web context. In fact, the Web was constantly being populated of

more and more resources and there was no chance of monitoring and keeping track of all the changes and modify the ranking functions accordingly in order to satisfy all the possible queries.

In that period Learning to Rank [12] became a viable alternative to the previous approach. Learning to Rank consists in letting the model itself learn which are the best parameters in order to produce the best ranking lists possible. This was possible thanks to the many Query Logs that started being available due to the many users surfing the Web. In this way, it is possible to constantly update the model's ranking functions based on the Query Logs that are being produced by users every time they are issuing a search query.

Nowadays Information Retrieval systems are employing advanced techniques in order to be as efficient and effective as possible. In particular, thanks to the development of new Machine Learning ideas, now it is possible to apply Natural Language Processing methods and Deep Learning models to better satisfy the users' needs. In fact, Machine Learning and Artificial Intelligence has become more and more pervasive in this field allowing systems to be able to provide relevant results to almost every possible query.

Last research frontiers are trying to improve the current state-of-the-art solutions in some specific fields such as:

- **Conversational Search** : humans are engaging conversations with IR systems. The objective of these systems is to capture even the voice tone and the conversation contexts in order to provide the best results possible to the users in a conversational way [13]. Practical examples of Conversational Search IR systems are the vocal assistants that we have in our smartphones or in our homes.
- **Quantum IR**: this field consists in exploiting the concepts of Quantum Mechanics to formulate IR models and problems [14]. This does not deal with the actual implementation of Quantum algorithms to tackle specific problems but rather in reformulating the considered problems according to the principles and formalisms of Quantum Mechanics.

2.1.2 EVALUATION OF IR SYSTEMS

Evaluation of IR systems can be very complex because effectiveness is a qualitative and subjective concept.

2.1. INFORMATION RETRIEVAL

To better explain this fact, please consider the following simple example.

Consider 2 students trying to improve their grades in a Physics class. Student A decides to study for several hours every day memorizing formulas and practicing problems until the student has understood them thoroughly. Student B instead decides to attend a weekly tutoring session, work through practice problems with classmates and focus on understanding the overall picture rather than memorizing formulas.

For Student A, the daily intensive studying may be very effective in helping him/her achieve a high score on his tests and assignments. Since Student A is satisfied with the results, he/she considers his/her approach to be effective.

On the other hand, Student B may find that the interactive and collaborative approach to learning works better. In fact it may allow him/her to have a deeper understanding of the material, feel more engaged in class and perform well on exams. So Student B may consider his/her approach to be more effective.

Imagine that now both of them asks to an IR system which technique they should be following to have the best results in the next exam. If the IR system suggests the technique employed by Student A, then Student A will think that the IR system is working very well but Student B might be disappointed with the provided answer. A similar but opposite situation would happen if the system suggests the method employed by Student B.

As you can notice with this very simple example, we could claim that our system is performing well in terms of effectiveness since it is providing good answers to the requests made by the 2 Students. But from the singular students' perspectives this is not the case.

It's clear that evaluation of IR systems can be a rather complex task, as relevance is a subjective concept and different users may have different opinions on what constitutes relevant information. To overcome this problem, some evaluation methods employ relevance judgments from multiple users, such as through crowd sourcing or evaluation campaigns, to estimate the relevance of documents following the Cranfield Paradigm [15]. Evaluation campaigns will be discussed more in depth in Section 2.3.

After having understood that the evaluation process of an Information Retrieval system can be quite complex, we can now have a look at some useful evaluation measures that are usually employed to establish if a system is performing well or not in terms of effectiveness. This will allow us to be able to

better interpret how the analysis of Quantum IR systems can be carried out.

We can identify mainly 2 families of evaluation measures: Set-based Evaluation Measures and Rank-based evaluation measures. The difference between these 2 families is that Rank-based evaluation measures are considering the ranked lists of documents, therefore in that case it is not only important whether a document has been retrieved or not but also the position in the ranked list. In the Set-based evaluation measure we can mainly find Precision, Recall and F-measure.

Precision and Recall are 2 evaluation measures that are commonly employed also in other fields. In the field of IR, Precision measures the proportion of relevant documents retrieved among all the documents retrieved, while Recall measures the proportion of relevant documents retrieved among all the relevant documents in the collection [16]. We can formulate the Precision P and the Recall R as

$$P = \frac{|relevant\ documents| \cap |retrieved\ documents|}{|retrieved\ documents|}, \quad (2.1)$$

$$R = \frac{|relevant\ documents| \cap |retrieved\ documents|}{|relevant\ documents|}. \quad (2.2)$$

F-measure is an evaluation measures that combines Precision and Recall into a single value, balancing the trade-off between them [17]. We can formulate the F-measure F as

$$F = \frac{2}{\frac{1}{P} + \frac{1}{R}} = 2 \cdot \frac{P \cdot R}{P + R}. \quad (2.3)$$

In the Ranked-based evaluation measures family we can mainly find instead Precision and Recall at Document Cut-off, Average Precision and Discounted Cumulated Gain.

Precision at Document Cut-off and **Recall at Document Cut-off** are related to Precision and Recall. In this case, we are considering that our system retrieves a ranked list of documents and we want to measure Precision and Recall considering only part of the total retrieved documents. Precision at Document Cut-off represents the proportion of relevant documents among the top k retrieved documents and the Recall at Document Cut-off represents the proportion of relevant documents considering only the top k retrieved documents among all

2.1. INFORMATION RETRIEVAL

the relevant documents in the collection. k is the cut-off point. We define r_n as

$$r_n = \begin{cases} 1 & \text{if document } n \text{ is relevant} \\ 0 & \text{if document } n \text{ is not relevant} \end{cases} \quad (2.4)$$

RB as the total number of relevant documents, then we can formulate the Precision at k $P(k)$ and the Recall at k $R(k)$ as

$$P(k) = \frac{1}{k} \cdot \sum_{n=1}^k r_n, \quad (2.5)$$

$$R(k) = \frac{1}{RB} \cdot \sum_{n=1}^k r_n. \quad (2.6)$$

Average Precision is the average of the Precision values computed at each relevant item retrieved [18]. A high Average Precision value indicates that the considered system is able to retrieve a high number of relevant items with high precision, while a low Average Precision value indicates that the retrieval system is not able to retrieve many relevant items or it is retrieving many irrelevant items. After having defined R as the set of relevant documents, then we define the Average Precision AP as

$$AP = \frac{1}{RB} \cdot \sum_{k \in R} P(k). \quad (2.7)$$

Discounted Cumulative Gain measures the usefulness, or gain, of a ranked list of items. Each item's gain is discounted based on its rank, so that items appearing earlier in the ranking receive a higher weight in the calculation of the Discounted Cumulative Gain. The discount factor ensures that the relative importance of the items decreases logarithmically as their rank decreases. Discounted Cumulative Gain is very important since it allows to account also for multigrade relevance, which means that a document can have several relevance levels (e.g. not relevant, partially relevant, completely relevant etc...). After having defined r_k as the relevance value of the document, we define the Discounted Cumulative Gain DCG as

$$DCG(k) = \sum_{n=1}^k \frac{r_k}{\max(1, \log_b k)} \quad (2.8)$$

where b is an hyperparameter that indicates the patience of a user in scanning the result list: the higher b the more patient a user is.

Usually we also consider the normalized version of the Discounted Cumulative Gain as $nDCG$, which is defined as

$$nDCG(k) = \frac{DCG(k)}{iDCG(k)} \quad (2.9)$$

where $iDCG(k)$ is the ideal DCG .

2.1.3 APPLICATIONS OF IR SYSTEMS

Information Retrieval systems are nowadays becoming more and more pervasive in several fields. In fact, the increasing volume of digital information has created a growing demand for Information Retrieval systems that can effectively manage and make sense of large amounts of data.

The most common application of IR systems is in Search Engines. A Search Engine is a software application designed to help people find information on the Web [19]. It works by using algorithms to search through a very large collection of websites and web pages, and then return the most relevant results based on the user's search query.

There have been studies providing some estimations about the number of indexed web pages according to the Google Search Engine. As of today, there are almost 50 billions indexed web pages considering the Google's index only [20]. It is very difficult to provide an exact number of indexed Web pages since the web is constantly changing especially in this period where Internet is becoming pervasive and fundamental in our daily routine.

As you can imagine, having to deal with this large amount of data requires to build very complex systems that are distributed all over the world to satisfy the users' needs.

Another very important application of IR systems is for Product Search. Since the rise of large internet-based e-commerce sites, the growth of online shopping has been exponential, with a huge increase in the number of consumers choosing to buy online rather than in-store [21].

In fact there are several advantages in doing shopping online, such as getting

2.1. INFORMATION RETRIEVAL

products delivered straight to your home, buying products in just a couple of clicks and finding whatever needed thanks to the vast catalog of available products possibly with cheaper prices with respect to the ones that can be found in physical stores.

Product Search is fundamental for online shopping because it allows customers to find certain products that satisfy a set of criteria based also on their preferences.

In Product Search, a customer usually provides the name of the needed good, usually with the possibility of specifying a set of attributes to restrict the search scope only on a set of possible items with the given characteristics. The system will then search according to the specified characteristics and will retrieve the best matching products in the catalogue.

Media Search [22] (or Multimedia Search) is also a popular application nowadays, due to the proliferation of personal multimedia devices.

These systems need to extract useful data from audios, images or videos in order to satisfy specific needs.

We consider as an example 2 famous applications that almost everyone is aware of: Spotify and Shazam.

In Spotify it is very important to have a Recommendation System capable of suggesting songs tailored to users' preferences. In this way, any user will have a system capable of suggesting tracks based on what the user had listened to before, what music genre the user prefers and which musical groups the user likes the most.

Obviously, the underneath algorithm is way more complex and probably considering several different features to predict the best songs for each users but this should give an idea of how Recommendation Systems are actually very useful.

On the other hand, Shazam is an application that can be used to analyze a song that is currently being played and retrieve its title, artist, album and URL pointing to the given song.

As you can imagine, Machine Learning plays a fundamental role into capturing the important features of a song which will be used to search through the collection of all the audio tracks in order to return the song that best matches the one provided as input by the user.

2.2 QUANTUM COMPUTING

Quantum Computing (QC) is a type of computing that uses quantum-mechanical phenomena to perform operations on data.

Unlike classical computers, which use bits to represent information, quantum computers use instead quantum bits, or qubits.

The unique properties of qubits allow quantum computers to perform certain types of computations much faster than classical computers, thus bringing much attention to Quantum Computing because of its potential benefits with respect to solutions involving classical hardware. In fact, Quantum Computing is nowadays a deeply researched field because of the many different areas where it can be applied.

It is worth mentioning that QC devices are still in their early stages of development but, thanks to the research, there has been a huge progress that has led to the implementation of more and more powerful quantum computers that are currently able to tackle even some practical and realistic problems as we will see in the next sections.

2.2.1 INTRODUCTION TO COMPUTATIONAL COMPLEXITY

We are usually interested into estimating the amount of resources used by the computer to understand the complexity of the computation. Resources can be either the *time* required to perform the given computation or the *space* which is related to the amount of memory employed.

Generally we are not interested in the exact amount of these employed resources to perform an algorithm but rather to their estimation because the amount of resources used depends also on the physical architecture underneath [23].

This is achieved by means of the so called **Big O Notation**, which is used to classify algorithms according to how their run time or space requirements grow as the input size grows.

We can formally define it in the following way:

considering the functions $f(n)$ and $g(n)$, then $f(n) = O(g(n))$ if \exists constants $c > 0, N > 0$ such that $0 \leq f(n) \leq c \cdot g(n) \forall n \geq N$.

To put it in simple words, O describes the **upper bound** of the complexity.

2.2. QUANTUM COMPUTING

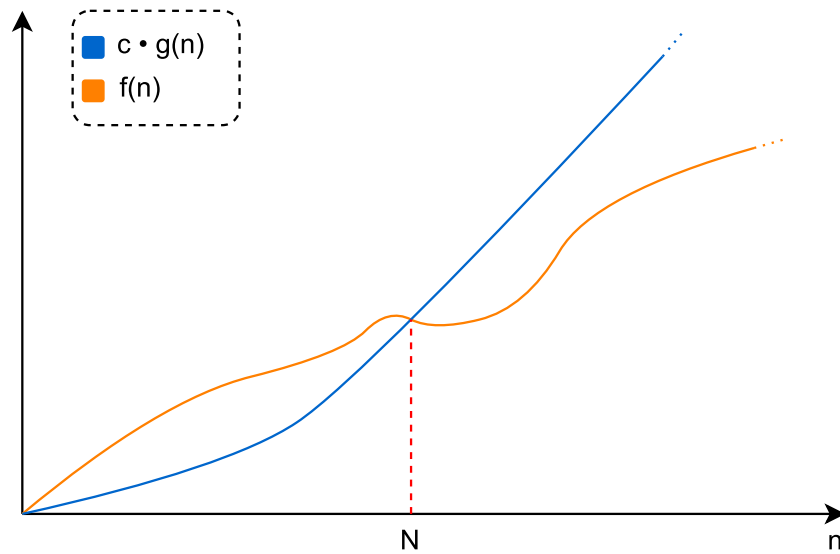


Figure 2.3: Graphical representation where $f(n) = O(g(n))$.

Usually we consider an algorithm to be efficient with respect to some resource if the amount of resources used by the algorithm is at most *polynomial* ($O(n^k)$ for some k). In general, if possible, we prefer our algorithms to be *linear* ($O(n)$) or *logarithmic* ($O(\log n)$) in the amount of resources used.

Unfortunately there are algorithms which require an amount of resources that is *exponential* which means $\Omega(c^n)$ for some c , where Ω describes the lower bound of the complexity.

To better understand why efficiency is fundamental, we provide here a table that represents an example of the amount of time required by algorithms having different complexities with respect to different input sizes.

Complexity	$n = 10$	$n = 20$	$n = 40$	$n = 60$
linear: n	10 μs	20 μs	40 μs	60 μs
polynomial: n^2	100 μs	400 μs	1.6 ms	3.6 ms
polynomial: n^5	0.1 s	3.2 s	1.7 min	13.0 min
exponential: 2^n	1 ms	1 s	12.7 days	366 centuries

Table 2.1: Table representing the time required based on complexity.

As you can clearly see from this simple example, even for relatively small sizes

of our input n , if the algorithm requires an exponential time to be executed then it is impossible to solve the problem.

According to the Automata theory and, in particular, to the Church-Turing Thesis, a problem can be solved on any computer if and only if it can be solved on a Turing Machine. The Turing machine is a computing model consisting of a finite set of states, an infinite tape where symbols from a finite alphabet can be read or written by means of a moving head and a transition function that returns the next state in terms of the current state and the current symbol pointed to by the head.

This tells us that a problem can be solved on a modern computer if we can simulate the execution of the algorithm employed to solve the problem on a Turing machine. Turing machines are usually employed to study computability properties such as undecidability and intractability.

We can then define a Nondeterministic version of the Turing machine (also called Probabilistic Turing machine) as a Turing machine that is capable of making a random choice at each step. It has been proven that for each Nondeterministic Turing machine, there exists a Deterministic Turing machine such that they both accept the same language, which means that they can be employed to solve the same problem [24].

This has a caveat: the amount of time required to solve the problem by the Deterministic Turing machine is exponentially larger with respect to the Nondeterministic Turing machine. In fact, there are some important problems that we know how to solve efficiently with a Nondeterministic Turing machine but not with a Deterministic Turing machine.

We can then define NP as the set of decision problems solvable in polynomial time by a Non-deterministic Turing machine but not by a Deterministic Turing machine. It's worth noting that the $P = NP$ problem asks whether NP problems are in fact solvable in polynomial time, and remains one of the most important open problems in theoretical Computer Science. It is one of the seven Millennium Prize Problems which are seven well-known complex mathematical problems selected by the Clay Mathematics Institute in 2000 [25]. The Clay Institute will award 1 million dollars for the first correct solution for each one of these seven problems.

We mention here 2 other fundamental classes of problems that are:

2.2. QUANTUM COMPUTING

- *NP-hard* problems: *NP-hard* problems are a subset of *NP* problems that are at least as difficult to solve as the hardest *NP* problems. In other words, if an algorithm can efficiently solve an *NP-hard* problem, it can efficiently solve all *NP* problems. However, the reverse is not necessarily true: an algorithm that can efficiently solve an *NP* problem may not be able to efficiently solve an *NP-hard* problem.
- *NP-complete* problems: *NP-complete* problems are a subset of *NP-hard* problems that have the additional property that, if an algorithm can solve one *NP-complete* problem in polynomial time, it can solve all *NP* problems in polynomial time. In other words, *NP-complete* problems are the "hardest" problems within *NP*.

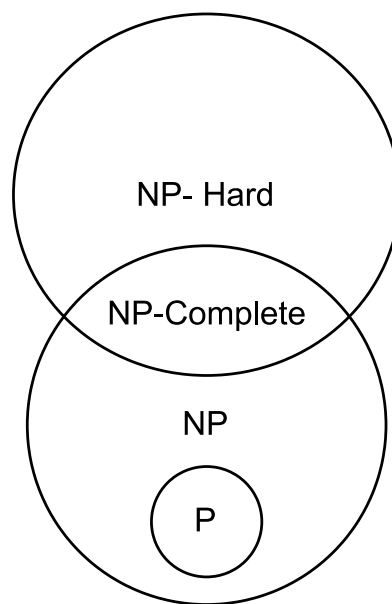


Figure 2.4: Graphical representation of the classes of problems.

QC has the potential to solve certain *NP-hard* problems more efficiently than classical computing methods. However, it's important to note that not all *NP-hard* problems can be solved efficiently on a quantum computer.

One famous example of an *NP-hard* problem that can be solved more efficiently on a quantum computer is the problem of factoring large numbers, which is a key step in many public-key cryptography algorithms. Shor's algorithm [26], which is a quantum algorithm for factoring, has been shown to be exponentially faster than the best known classical algorithms for this problem.

However, it's also worth noting that while Quantum Computing can provide exponential speedup for certain problems, it's still an area of active research and there is much that is not yet understood about the potential and limitations of

Quantum Computing for solving *NP*-hard problems. Furthermore, the practical implementation of quantum algorithms can be challenging and the required resources (e.g. the number of qubits and the error-correction overheads) can be substantial. In fact, Quantum Computing systems are very fragile and require particular attention to handle their possible errors.

2.2.2 MAIN CONCEPTS OF QUANTUM COMPUTING

Quantum Computing deals with modern physics that explains the behavior of matter and energy of an atomic and subatomic level. Quantum Computing makes use of quantum phenomena, such as quantum bits (known as qubits), superposition and entanglement to perform operations [27].

SUPERPOSITION

Superposition in Quantum Computing refers to the fact that a qubit can be in multiple states simultaneously. While a normal bit can only be in state 0 or 1, a qubit can be in a superposition of states 0 and 1 simultaneously.

To better grasp this concept, we can relate it to classical physics considering sound waves. Each wave can be seen as a combination of several waves having different frequencies, in a sort of superposition.

There is nevertheless a fundamental difference between Quantum superposition and superposing classical waves. A quantum computer consisting of n qubits can exist in a superposition of 2^n possible states, while playing n musical sounds with all different frequencies, can only give a superposition of n frequencies. Therefore, adding classical waves scales linear, where the superposition of quantum states is exponential [28].

This counterintuitive phenomenon allows Quantum Computers to calculate a wide variety of possible outcomes simultaneously using several qubits in superposition. The final result of a calculation emerges only once the qubits are measured, which immediately causes their quantum state to "collapse" to either 1 or 0.

Superposition has been observed and explained in the Double-Slit Experiment [29]. In this experiment it has been shown that light behaves like particles when sensors are employed to capture the route taken by the light while light behaves like waves if these sensors are not employed. This experiment demonstrated both the duality of photons and the concept of superposition that is employed

2.2. QUANTUM COMPUTING

in Quantum Computers by means of qubits.

ENTANGLEMENT

Entanglement means that the state of one qubit can be correlated with the state of another qubit. Changing the state of one of the qubits will instantaneously change the state of the other one in a predictable way. This happens even if they are separated by exceptionally long distances.

As you entangle more and more qubits together, the ability of the system to make calculations grows not in a linear fashion, but exponentially.

2.2.3 QUANTUM ANNEALING

In the latest years, researchers managed to build different types of quantum computers that can be used to tackle different kind of problems.

Mainly we can find gate models and quantum annealers.

A gate model quantum computer is a universal quantum device, which means it can perform any quantum operation that can be represented as a sequence of quantum gates where a quantum gate is a basic quantum circuit which operates on a small number of qubits. This makes gate model quantum computers very flexible and powerful, but also more complex and harder to control.

On the other hand, a quantum annealer is a specialized quantum device that is designed to solve a specific type of problems known as optimization problems [30]. Optimization problems are problems in which we want to find which is the optimal solution among all the possible feasible solutions to a given problem.

In a gate model quantum computer, quantum states are typically represented using the quantum circuit model, where quantum states are represented as quantum circuits made up of quantum gates. This means that a quantum state in a gate model quantum computer can be represented using a sequence of quantum gates acting on some initial state.

In contrast, in a quantum annealer, quantum states are represented using the quantum adiabatic model, where quantum states are represented as the ground state of a time-dependent Hamiltonian. This means that a quantum state in a quantum annealer can be represented as the lowest-energy state of a Hamiltonian that depends on time.

To sum up, we can represent the main differences between the 2 systems as follows.

In Quantum Annealing, you are harnessing the natural evolution of quantum states, although that evolution cannot be controlled. The problem is in fact set up at the beginning and then you let quantum physics do its natural evolution without using sensors to measure the process itself. The configuration at the end corresponds to the answer you were looking for.

In gate model Quantum Computing the aim is instead much more ambitious. What you are basically trying to do is being able to control and manipulate the evolution of that quantum state over time in order to perform specific operations. As quantum systems are highly delicate, this is indeed a very challenging task. However having that amount of control means that you can solve a bigger class of problems.

As a result of these differences, researchers have been able to scale Quantum Annealing devices up to thousands of qubits whereas gate model quantum computing currently only supports around 100 qubits. In fact, it is technically much more difficult to get the qubits to work together coherently in the gate models with respect to quantum annealing models.

In this work we will focus our attention on Quantum Annealers.

Quantum Annealing processors naturally return low-energy solutions; some applications require the real minimum energy (optimization problems) and others require good low-energy samples (probabilistic sampling problems) [31].

In an optimization problem, you search for the best of many possible combinations. An example of optimization problem is the Travelling Salesman Problem (TSP). In this problem we have a list of cities and the distances between each pair of cities. We then want to discover what is the shortest possible route that visits each city exactly once and returns to the origin city. This is a very famous *NP*-hard problem. The TSP naturally arises in many transportation and logistics applications, for example the problem of arranging school bus routes to pick up the children in a school district or finding the best route to deliver packages to customers.

These problems can be formulated as energy minimization problems. In fact, we can exploit the fact that in physics everything tends to seek a minimum energy state. This is also explained in physics as the principle of minimum energy.

One example of everything tending towards a minimum energy state can be

2.2. QUANTUM COMPUTING

seen in the behavior of a hot object cooling down to the temperature of its surroundings. When the temperature of an object is higher than its surroundings, it has more thermal energy and is therefore in a higher energy state. As it loses thermal energy to its surroundings, its temperature decreases and it tends towards a minimum energy state, which is the state of thermal equilibrium with its surroundings.

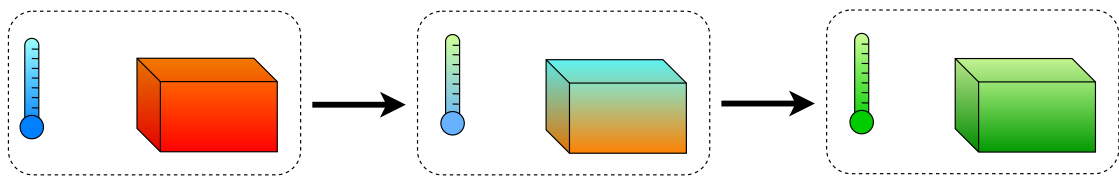


Figure 2.5: Representation of thermal cooldown of an object with respect to its environment.

Another example can be seen in the behavior of a pendulum. When a pendulum is released from a starting position, it swings back and forth. Over time, its swings become smaller and smaller because of air friction until it eventually comes to rest at its lowest energy state, which is its center of gravity. This is because the energy the pendulum had when it was released is converted into kinetic energy that is in part dissipated and then back into potential energy, until eventually all its energy is in the form of potential energy, and it comes to rest.

This is due to the fact that if it happens for a system to get to a lower energy state then it has no way to go back up without external energy because of the principle of conservation of energy.

This behavior is also true in the world of quantum physics. Therefore, Quantum Annealing simply uses quantum physics to find low-energy states of a problem and therefore the optimal or near-optimal combination of elements considering the given problem.

Quantum Annealing is also useful for Sampling Problems. In fact, sampling from many low-energy states and characterizing the shape of the energy landscape is useful for Machine Learning problems where you want to build a probabilistic model of reality. The samples give you information about the model state for a given set of parameters, which can then be used to improve the model.

Probabilistic models explicitly handle uncertainty by accounting for gaps in knowledge and errors in data sources. In fact, it is almost impossible to have datasets that do not contain errors or outliers. The distribution of the data is approximated based on a finite set of samples. In fact, we expect that if we have at our disposition enough samples then it will be possible for our model to generalize well and understand the correct true distribution of our data.

If the training process is successful, the learned distribution resembles the true distribution that generated the data, allowing predictions to be made on unobserved data. In this case we say that the model is able to generalize well.

For example, there are generative models that can be trained to generate samples resembling the input data distribution. Those include, but are not limited to, Generative Adversarial Networks [32] and Variational Autoencoders [33].

These Machine Learning models have been used to deal with images in various tasks such as super resolution [34]. It consists in reconstructing original images from their corresponding downsampled versions and is a practical application that allows to build models capable of improving the image quality.

We now provide an overview of how a quantum annealer works considering the ones provided by D-Wave, one of the biggest company in the Quantum field. The D-Wave devices will be then used in this project in our experimental phase and in the future QuantumCLEF evaluation campaign.

In a D-Wave quantum annealer a qubit's state is implemented as a circulating current, circulating clockwise for 0 and counter clockwise for 1. Following the Biot–Savart law, a magnetic field is produced with a specific direction based on the direction towards which the current circulates.

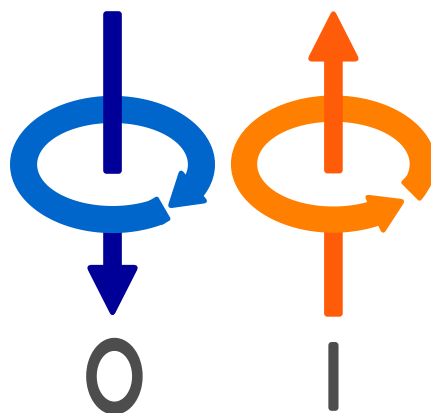


Figure 2.6: Representation of how a qubit's state is implemented in a D-Wave quantum annealer.

2.2. QUANTUM COMPUTING

In Quantum Annealing the qubits start from a superposition state and finally they will reach a state which will be either 0 or 1 at the end of the Annealing process.

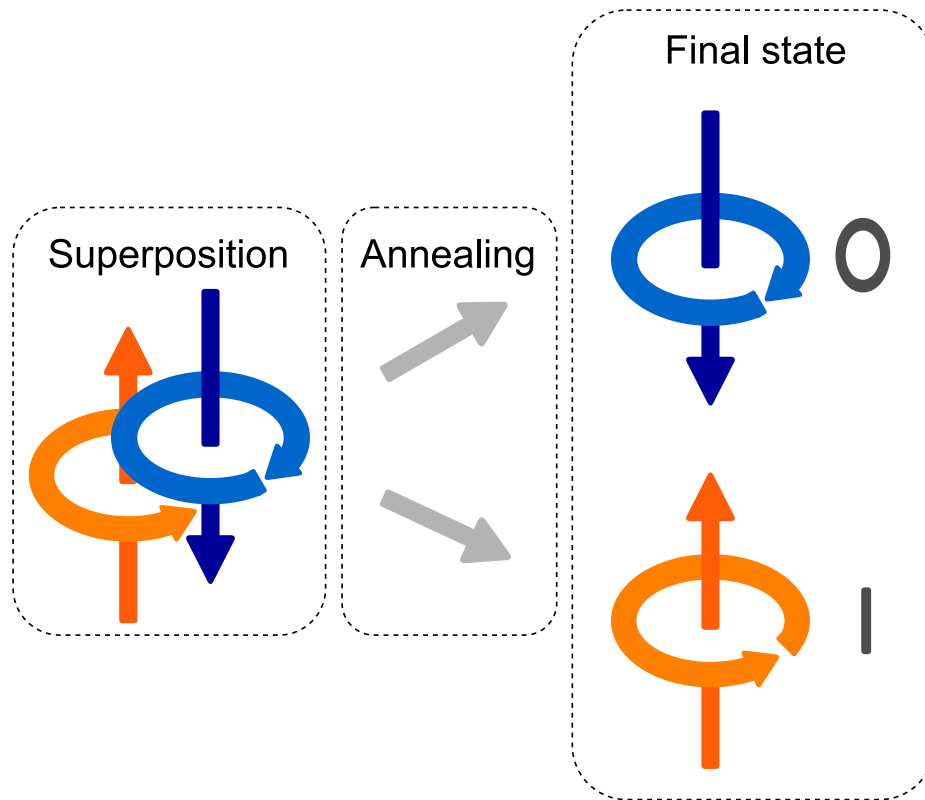


Figure 2.7: A very simple representation of the Annealing process involving a single qubit.

As we already mentioned, the Annealing process tends to seek low energy values based on the states of the qubits.

We can visualize this physical process by means of an energy diagram that is presented below. In this case we are considering a simple case where there is only 1 qubit [31].

At the beginning there is just one valley and the lowest point corresponds with the superposition state of the considered qubit. When the Quantum Annealing process is executed, a barrier is raised and this turns the energy diagram into a double well potential. We can see that the low point of the left valley corresponds to the 0-state while the low point of the right valley corresponds to the 1-state.

In this case the probability of the qubit ending in one of the two considered

states is equal, namely 50% to end up in any of the 2 possible states.

There is the possibility to introduce a *bias* that will affect the probability of the qubit to end up in one state rather than the other one. This can be achieved by applying an external magnetic field to the qubit with a specific strength which can be chosen according to the constraints of the considered problem.

In this case the energy diagram gets modified and there will be a higher probability for the qubit to end up in the 1-state rather than the 0-state following the principle of minimum energy that we mentioned before.

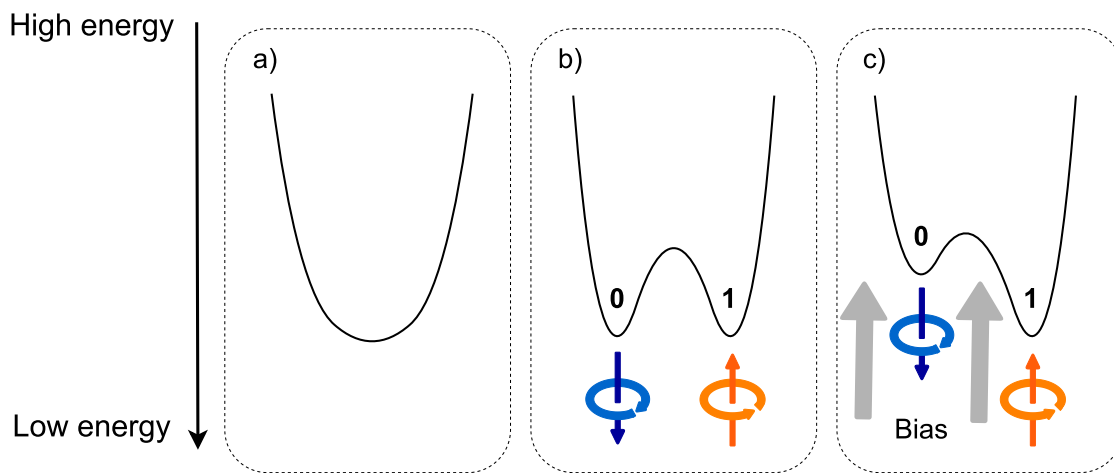


Figure 2.8: Energy landscape affected by bias.

To exploit the entanglement phenomenon, some devices called *couplers* are being employed.

A coupler basically defines how qubits influence each other. In fact, through a coupler, it is possible to make two qubits tend to end up in the same state or to make them tend to be in opposite states.

It is important to understand that the coupled qubits now must to be considered as a single object that can be in several possible different states simultaneously, which are the combination of states of the considered coupled qubits.

As it happens for the qubit bias, it is also possible to set another parameter, called *strength*, between each coupled qubits that represents how strong the correlation between them is.

2.2. QUANTUM COMPUTING

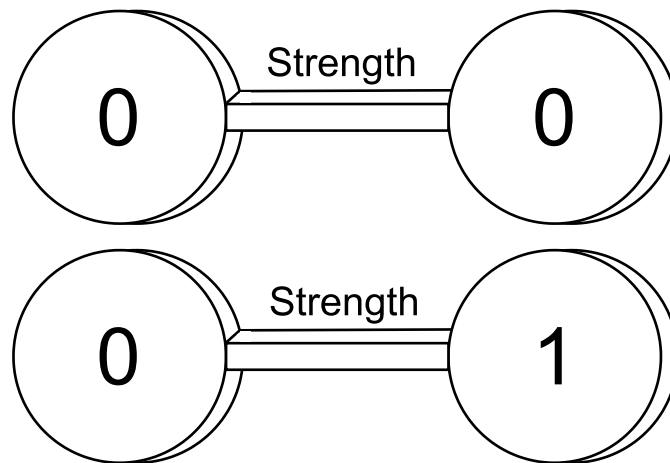


Figure 2.9: Representation of couplers.

We can make a very simple example to visualize what happens in case we have 2 coupled qubits, thus having 4 possible different states that are (0,0), (0,1), (1,0) and (1,1) [31].

When formulating a problem, users choose values for the biases and couplers. The biases and couplings define an energy landscape, and the D-Wave quantum computer finds the minimum energy of that landscape.

As you might notice, the number of different possible energy states in the energy landscape grows exponentially with the number of qubits employed! This is a particularly important feature of quantum annealers that makes them much more powerful compared to classical machines.

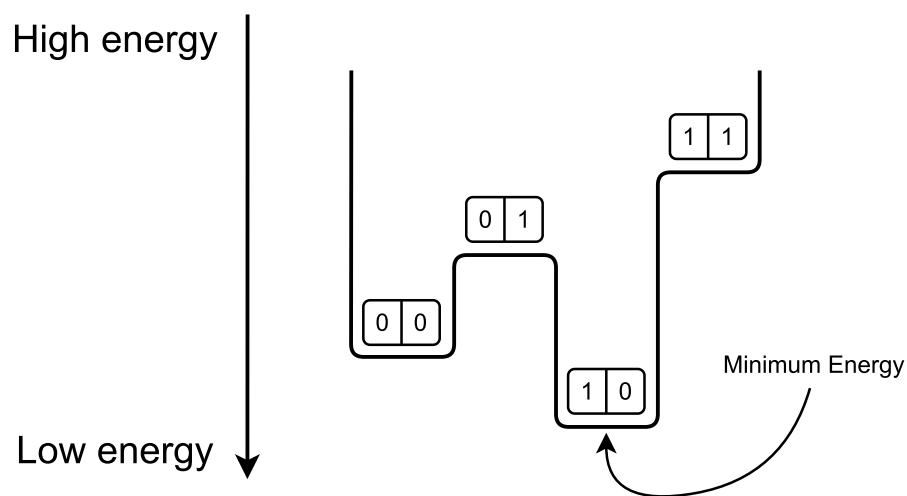


Figure 2.10: An example of Quantum Annealing considering 2 qubits coupled together.

In summary, the system starts with a predetermined set of qubits. Each qubit is initially in a superposition state of 0 and 1 and they are not yet coupled. When the Quantum Annealing process starts, the couplers and biases are introduced and the qubits become entangled according to the specifications provided by the user. At this point, the system is in an entangled state of many possible answers. By the end of the annealing process, each qubit is instead in a classical state (either 0 or 1) that represents the minimum energy state of the problem, or one very close to it [31].

Since we might not find the best solution in just one attempt, usually we run this process several times and pick the best solution obtained among all the runs. However this is not a problem in terms of efficiency since even the annealing process happens in a matter of microseconds.

2.2.4 QUANTUM ANNEALING: ENVIRONMENTAL CONSIDERATIONS

Quantum Annealing, like other Quantum Computing technologies, has the potential to be more energy-efficient than classical computing in certain applications. However, it is difficult to make a direct comparison between Quantum Annealing and classical computing in terms of environmental impact, as the two technologies have different strengths and limitations.

One of the advantages of Quantum Annealing is that it can solve certain optimization problems more quickly than classical computers, which can lead to energy savings in some cases. For example, some problems that would require an impractically long time to solve on a classical computer could be solved much more quickly on a quantum annealer, potentially reducing the amount of energy required to find a solution. However, quantum annealers are currently limited in their capabilities and may not be able to solve all optimization problems efficiently.

On the other hand, Quantum Computing technology is still in its early stages of development, and the energy consumption of quantum annealers can vary widely depending on the specific hardware and algorithms being used. Additionally, quantum annealers require specialized cooling systems to maintain the low temperatures required for quantum operations, which can be energy-intensive. In fact, the quantum annealers provided by D-Wave require to be working at temperatures close to the absolute zero (generally lower than 0.05

2.2. QUANTUM COMPUTING

Kelvin; the new systems are working at temperatures lower than 0.015 Kelvin) to avoid having environmental interferences and to and improve the coherence of the qubits. The refrigeration system employed is called dilution refrigerator. It has been made a comparison between the D-Wave Two quantum annealer and the NVIDIA DGX-1 server based solely on FLOP rate and computation time [35]. It has been seen that the D-Wave Two quantum annealer is equivalent to roughly 500 NVIDIA DGX-1 servers. When it comes to power draw, the D-Wave system's nominal power consumption is 16kW while 500 NVIDIA DGX-1 servers would require 1.6 MW; therefore Quantum Annealing, in principle, can consume far less power.

Anyway classical computing has a well-established infrastructure and a wide range of applications, and energy-efficient computing technologies have been developed and implemented over the years. Through classical hardware it is possible to solve a wider range of problems because Quantum Computing it is in its early stages of developments. However, classical computing can also consume a significant amount of energy, especially in data centers and high-performance computing applications.

Overall, it is difficult to make a definitive statement on whether Quantum Annealing is more environmentally friendly than classical computing in this moment. The environmental impact of each technology depends on a variety of factors, including the specific application, the hardware and algorithms used and the energy sources powering the computing systems.

It is likely that in the future Quantum Computing devices can provide benefits with respect to classical computing devices considering environmental sustainability. In fact, since the first generation of D-Wave systems was introduced in 2011, the amount of power drawn was mainly due to the refrigeration and has remained constant. This trend is expected to continue as computational power grows with successive generations of QPUs [35]. This will allow to have more powerful devices requiring almost the same amount of energy that was being used so far, thus improving the performance per watt.

2.2.5 APPLICATIONS OF QUANTUM COMPUTING

Quantum Computing can be applied in several fields to improve the current solutions especially in terms of efficiency. In fact, as we saw before, Quantum

Computing has the potential to solve very complex problems in a very short amount of time.

An area where quantum holds promises is in material discovery and drug development [36]. In fact, developing new useful molecules requires combinatorics because there are many possible combinations of atoms and many possible ways that they can bond.

There are classes of molecules that are too challenging to simulate with classical approaches because of the underlying combinatorics, but it will be possible to simulate them in a timely manner once quantum computers will be improved sufficiently.

Cybersecurity is another area where Quantum Computing can make the difference.

In fact, corporate data are frequently stored through encryption techniques to ensure security [37]. However, hacking attacks are getting more sophisticated, therefore many existing encryption methods are at risk.

In this scenario, Quantum Computing can be used to improve the performance of traditional file encryption and decryption algorithms.

It is expected to take existing information and communication technologies to new levels based on fast computing power and strong security.

Combinatorics challenges are also common in banking and finance, from arbitrage to credit scoring to derivatives development. One possible way in which banks and other financial institutions deal with these problems is to constrain them in order to make them more tractable, but constraining the set of possible solutions means that sometimes the best solution is never found.

There is a potential for quantum computers to shed insights into larger problems where constraints are relaxed and where more outcomes are possible.

A surprisingly large number of business problems can be framed as variations of the traveling salesman including circuit design, package delivery and train scheduling. More specifically, researchers have identified combinatorics problems in banking and finance that might benefit from quantum computing, including portfolio optimization, foreign exchange arbitrage and credit scoring [38].

2.3 EVALUATION CAMPAIGN

As already mentioned, evaluation campaigns play a fundamental role when it comes to the evaluation of IR systems. In fact, to establish whether a system is effective or not it is necessary to compare it against other possible approaches. In addition, it is impossible to assess how well performing an IR system is without experimental evaluation.

We refer to experimental evaluation as the assessment of the performance of a system by testing it against a dataset and provide results according to some common and known metrics.

Evaluation campaigns are also very important to compare several systems together allowing to understand if new ideas can improve the current solutions.

2.3.1 SHARED TASKS AND THEIR IMPORTANCE

Shared tasks can be seen as some sort of challenges in which many research groups participate by submitting their solutions to a given problem.

The goal of a shared task is to evaluate the performance of different methods for solving a given problem. This allows to promote research and development in the field exploring several solutions presented by research groups potentially having different backgrounds.

A shared task usually can take up to some months to be carried out. At the beginning, there is a call for applications so that groups are aware of the possibility of participating in the given task.

In most of the cases shared tasks are open to everyone, which means that anyone could participate. There are anyway cases in which tasks might have some requirements and thus participation is not granted to everyone.

After the group-registration phase, the research groups can proceed to work on the given problem by analyzing some provided baseline solutions and trying to improve their results according to their own ideas. In this part, groups are usually given some datasets and some specific constraints on how the results should be produced and the format of these results.

Finally, each group submits its final solution and, generally, a report or paper that describes how its solution works and performs against the given dataset.

After the submissions, judges will evaluate the solutions and the reports according to a pre-defined set of evaluation measures to assess whether some methods

are actually effective or not.

At the end, the judgements are provided so that the groups will be informed about how they performed in comparison to other groups. This is usually done by means of a ranking list that represents the score achieved by each group according to the evaluation metrics chosen.

This does not mean that this is necessarily a competition, but having ranking lists can spur groups to do the best they can in order to achieve the best results possible.

Shared tasks can be beneficial in many different ways, such as:

- Facilitating the comparison of different methods and techniques by providing a common evaluation framework.
- Encouraging collaboration and knowledge sharing among researchers by providing a forum for discussing and sharing results.
- Encouraging innovation and new ideas by providing a platform for researchers to showcase their work.

Shared tasks are usually organized by a group of researchers or a research organization, and can be sponsored by academic institutions, research funding agencies or industry partners.

2.3.2 CLEF

CLEF, which stands for Conference and Labs of the Evaluation Forum, is a research evaluation initiative that was established in 2000 as a spin-off of the TREC Cross-Language Track.

It focuses on stimulating research and innovation in multimodal and multilingual information access and retrieval [39].

Multimodality is intended as the ability to deal with information not only conveyed by multiple media, but also coming in different modalities such as the Web, social media, news streams, specific domains, and so on [40].

It is organized by a consortium of European research institutions and universities. It aims to promote research and development in the field of Information Retrieval. CLEF provides a common evaluation framework for researchers to evaluate their systems and methodologies in a variety of IR shared tasks.

2.3. EVALUATION CAMPAIGN

The CLEF initiative has organized through the years several Labs and Workshops. Benchmarking Labs provide a “campaign-style” evaluation for specific information access problems. The topics covered by campaign-style labs can be inspired by any information access-related domain or task. Workshops, instead, explore issues of evaluation methodology, metrics, processes etc. in information access and closely related fields. Here we report a list containing only some of the Labs and Workshops organized from 2010:

- Web People Search (WePS): a Workshop focused on person name ambiguity and person attribute extraction from Web pages and on online reputation management for organizations.
- Music Information Retrieval (MusiCLEF): a brainstorming Workshop promoting the development of new methodologies for music access and retrieval on real public music collections.
- Image Retrieval (ImageCLEF): its goal is to support multilingual users from a global community accessing an ever growing body of visual information.
- Intellectual Property in the Patent Domain (CLEF-IP): a Lab focused on various aspects of patent search and intellectual property search in a multilingual context.
- Biodiversity Identification and Prediction(LifeCLEF): a Lab which aims at boosting research on the identification and prediction of living organisms in order to solve the taxonomic gap and improve our knowledge of biodiversity.

Thanks to CLEF activities over the last two decades, it has been possible to create a considerable amount of valuable resources which is now extremely useful for many types of text processing and benchmarking activities related to the IR domain.

According to Google Scholar Metrics, CLEF is among the top 20 venues for the "Databases and Information Systems" area.

3

Shared Tasks: Overview

We have already seen that Quantum Annealing can provide benefits in case of optimization problems. In this chapter we will try to understand better how a problem needs to be formulated in order to be correctly processed and solved by a quantum annealer.

We will then have an overview of the Feature Selection problem, a well known *NP*-hard problem that has already been tackled by quantum annealers in a previous work [41]. This could be a starting task for the future QuantumCLEF evaluation campaign.

Finally we will try to understand which kind of specific problems can be solved by means of Quantum Annealing in the IR field. These problems will be furtherly discussed in the next chapters as possible tasks that can be proposed for the QuantumCLEF evaluation campaign.

3.1 QUBO FORMULATION

The Quadratic Unconstrained Binary Optimization (QUBO) model has become more and more relevant in recent years. In fact, it has been discovered that it is possible to solve many combinatorial optimization problems in a similar way by firstly transforming them into their QUBO versions [42].

In order to solve problems through quantum annealers, it is necessary to formulate them according to their corresponding Binary Quadratic Model (BQM) or, more specifically, to their QUBO or Hising version that are isomorphic expres-

3.1. QUBO FORMULATION

sions of the same problem.

In this work we will specifically analyze the problems solved in terms of their QUBO version.

We start by the formal definition of a general QUBO problem. We define y as

$$y = x^T Q x \quad (3.1)$$

where x is a vector of decision variables and Q is a matrix of constant values. Then the QUBO problem can be formulated as

$$QUBO := \min(y) \quad (3.2)$$

or

$$QUBO := \max(y) . \quad (3.3)$$

As you can clearly see, this formulation is an optimization problem in which we want to minimize or maximize the objective function y according to the values of the variables x .

We highlight the fact that QUBO problems belong to the class of *NP*-hard problems, so even famous commercial solvers like CPLEX [43] generally struggle to find good solutions even after days or weeks of computation.

Here we will make an example of a QUBO problem to make things a little bit more clear.

We consider here the Number Partitioning Problem, in which given a set of numbers $S = \{s_1, s_2, \dots, s_m\}$ we want to find 2 subsets $S_1, S_2 \in S$ such that $S_1 \cap S_2 = \emptyset$ and $\min(\sum_{s \in S_1} s - \sum_{s \in S_2} s)$. In words, the problem consists in finding 2 subsets of S such that the sum of the elements belonging to one subset is as close as possible as the sum of the elements belonging to the other subset.

To formulate it into a QUBO problem then we need to make some modifications. First of all we define our set of decision variables $x = [x_1, x_2, \dots, x_m]^T$ in which we have that

$$x_i = \begin{cases} 1 & \text{if } s_i \in S_1 \\ 0 & \text{if } s_i \notin S_1 \end{cases} . \quad (3.4)$$

Then in this case we can formulate the sum of the elements belonging to S_1 as $sum_1 = \sum_{i=1}^m s_i \cdot x_i$. As a consequence, we can formulate the sum of the elements

belonging to S_2 as $sum_2 = \sum_{i=1}^m s_i - \sum_{i=1}^m s_i \cdot x_i$.

Now as before we need to minimize the difference y that is

$$y = sum_1 - sum_2 = k - 2 \cdot \sum_{i=1}^m s_i \cdot x_i \quad (3.5)$$

but using the QUBO formulation. Notice that k represents just a constant in this case.

Now we can apply some further derivations to bring the problem closer to a QUBO formulation:

$$y^2 = k^2 + 4 \cdot \left(\left(\sum_{i=1}^m s_i \cdot x_i \right)^2 - k \cdot \left(\sum_{i=1}^m s_i \cdot x_i \right) \right) \quad (3.6a)$$

$$y^2 = k^2 + 4 \cdot \left(\left(\sum_{i=1}^m s_i^2 \cdot x_i^2 \right) + \left(\sum_{i,j} s_i \cdot s_j \cdot x_i \cdot x_j \right) - k \cdot \left(\sum_{i=1}^m s_i \cdot x_i \right) \right) \quad (3.6b)$$

$$y^2 = k^2 + 4 \cdot \left(\left(\sum_{i=1}^m x_i \cdot s_i \cdot (s_i - k) \cdot x_i \right) + \left(\sum_{i,j} s_i \cdot s_j \cdot x_i \cdot x_j \right) \right) \quad (3.6c)$$

Therefore, we can finally formulate the problem as:

$$\min_x \quad y^2 = k^2 + 4x^T Q x \quad (3.7a)$$

$$q_{ii} = s_i \cdot (s_i - k) \quad \forall i \in [1, m] \quad (3.7b)$$

$$q_{ij} = s_i \cdot s_j \quad \forall i \neq j \text{ and } i, j \in [1, m] \quad (3.7c)$$

As you can see now we are getting closer to the QUBO formulation. We only have to remove k and 4 from the formulation. This can be done since those are just constants that do not influence the results. In fact we only want to find the variables that are providing the minimum result possible according to our objective function.

Therefore, the final formulation will be

$$\min_x \quad y^2 = x^T Q x \quad (3.8a)$$

$$q_{ii} = s_i \cdot (s_i - k) \quad \forall i \in [1, m] \quad (3.8b)$$

$$q_{ij} = s_i \cdot s_j \quad \forall i \neq j \text{ and } i, j \in [1, m] \quad (3.8c)$$

which resembles the classical QUBO formulation.

3.1. QUBO FORMULATION

The solution provided as output if we provide this formulation to a Quantum Annealer will be a vector x' of binary values in which if the i -th component of the vector is 1 then the corresponding element s_i will belong to S_1 and vice versa.

The first step when dealing with a problem that needs to be solved through Quantum Annealers is therefore to convert the problem in its QUBO formulation.

Sometimes this can be straightforward, but this is not always the case. Anyway, we can make use of a simple table to help us converting some constraints into penalties in such a way that they will respect the QUBO formulation:

Classical Constraint	Equivalent Penalty
$x + y \leq 1$	$P \cdot (xy)$
$x + y \geq 1$	$P \cdot (1 - x - y + xy)$
$x + y = 1$	$P \cdot (1 - x - y + 2xy)$
$x \leq y$	$P \cdot (x - xy)$
$x = y$	$P \cdot (x + y - 2xy)$

Table 3.1: Table representing some useful conversions between constraints to corresponding penalties.

We want to highlight that P is a constant that should be *large enough*, which means that P itself should be chosen according to the given problem.

After having formulated the QUBO version of the considered problem, it is necessary to undergo through the *minor embedding* step. It is a fundamental phase that consists in ensuring that the problem will fit on the hardware that will be used to solve the problem.

In fact, each QPU has different hardware specifications such as a limited connectivity and number of qubits.

This phase is carried out, for example, by creating auxiliary variables that will inherit some of the connections of the original ones [44]. For example, the *D-Wave Advantage* quantum annealer has 5000 qubits at its disposition which are

connected in a sparse graph called Pegasus.

In this topology each qubit is connected to a maximum of 15 others forming a complex topology.

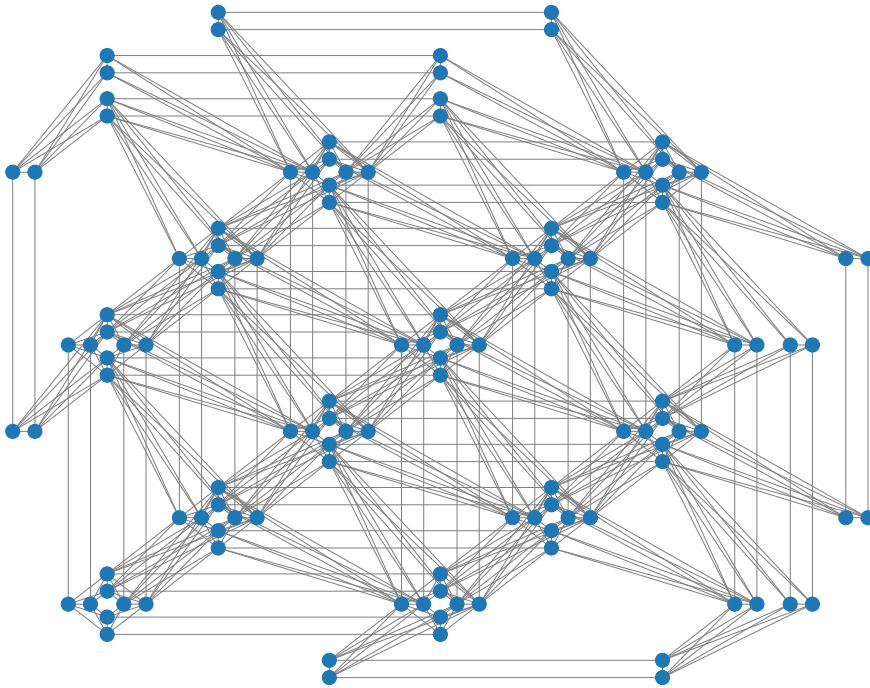


Figure 3.1: Portion of the Pegasus topology employed in the D-Wave Advantage QPUs. This figure has been obtained through D-Wave NetworkX Python library [45].

We provide here a simple example to better understand what happens in this phase.

Imagine having formulated a QUBO problem that can be represented through a graph that has a triangle topology but the QPU used requires us to represent it according to a square topology.

In this case we need to convert it to a square topology by adding an auxiliary variable. This practically means that we need to add a new node (namely a new qubit) to our triangular graph and a new edge connecting the new node to its corresponding old one to represent the same variable but according to the square topology.

3.2. FEATURE SELECTION

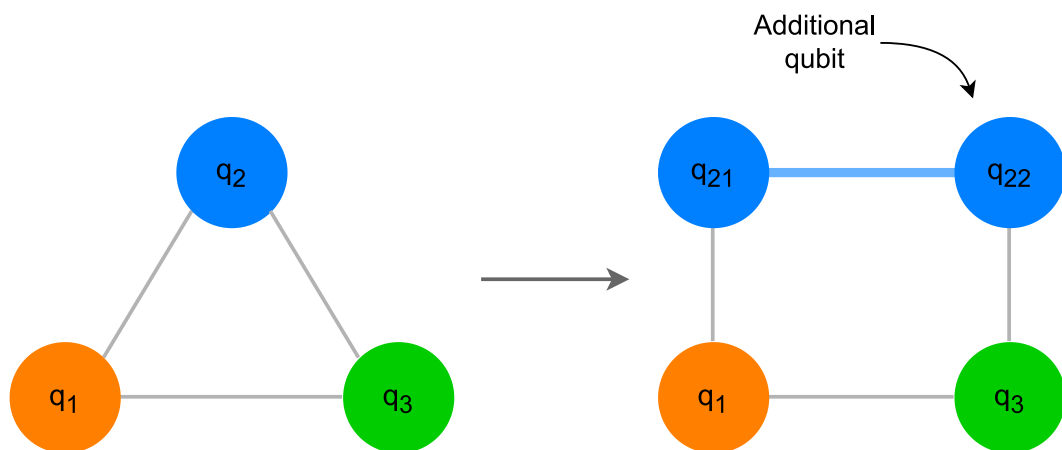


Figure 3.2: Representation of the process of minor embedding.

3.2 FEATURE SELECTION

Feature Selection is a well-known *NP*-hard problem that has been encountered in several fields such as Machine Learning and Information Retrieval. It involves identifying and selecting a subset of the most relevant features in a dataset to be used in building a model.

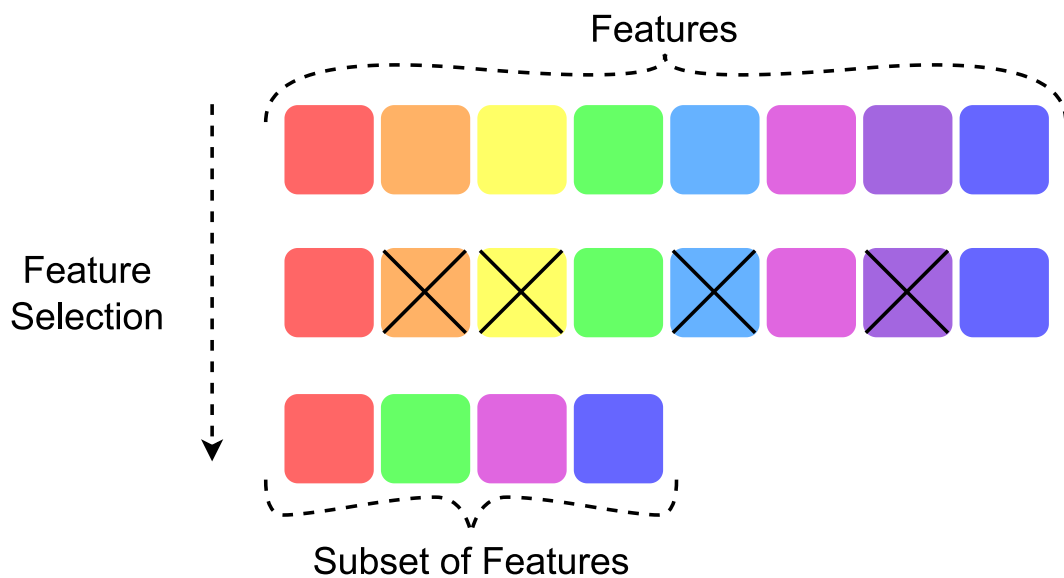


Figure 3.3: Graphical representation of the Feature Selection process.

Before formalizing the problem itself, we will consider some examples to allow you to better grasp why this problem is important to be solved.

Imagine that you have to train a Machine Learning model. As you probably know, training a Machine Learning model requires a Training dataset containing as many samples as possible. This is required in order to be representative of the input space from which we draw the Training samples. In this way, it will be possible to build a model that will be able to generalize well when trained appropriately, which means that it will be able to make correct predictions on unseen samples.

Training requires to provide the Training samples to the model itself usually in an iterative process. The model will then adjust its parameters in order to minimize a given loss function.

After having provided this very simple and brief overview of how training a model works, we can now consider a practical example.

Imagine that you have to build a model to predict whether a customer will buy a certain product or not in an e-commerce website. The dataset you have at your disposition contains samples having 1000 features each. For example we can think of a sample to be a vector $v \in \mathbb{R}^{1000}$ in which each component is a feature. We consider that each sample represents a customer with his or her corresponding data. We might have that feature v_0 represents how many products a customer has bought that are corresponding to the category *clothes*, v_1 represents how many products a customer has bought that are corresponding to the category *food* and so on.

Now, imagine that v_{100} represents how much time the given customer has spent on our website, then is it an important feature when it comes to predict whether a customer would buy a product or not?

Maybe having to deal with all the 1000 features is not the best situation possible for several reasons:

- Dealing with high dimensional vectors of features might require a more complex model that will take more time when it comes to make predictions, thus decreasing efficiency.
- Implementing a more complex model to handle the high dimensional training data can create some issues related to overfitting, a well-known problem in Machine Learning. The problem of overfitting happens when the model learns "by heart" the Training samples and is not able to generalize well on new data. We have also the opposite problem in which the model is too simple and is not able to predict well on our data: the problem of underfitting. We provide below a very clear and simple example of the problem of overfitting and underfitting.

3.2. FEATURE SELECTION

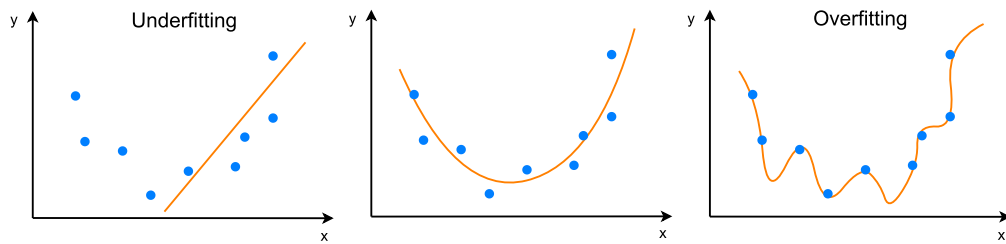


Figure 3.4: Example of underfitting and overfitting.

It is clear from the figure above that the best model is probably the one in the middle.

The model that is experiencing the overfitting issue (the one on the right) is performing perfectly on the training dataset but it is unstable. In fact, even for small changes on our training dataset it will not be able to correctly predict the corresponding labels.

Instead, the model experiencing the underfitting issue (the one on the left) is not able to perform well even on the training dataset.

As you can see, it is not guaranteed that the more complex the model is, the better its results. In fact, introducing several parameters into the model can cause the model to be able to overfit on the provided data due to its high level of complexity.

Here we mention a famous and funny quotation regarding overfitting:

John von Neumann: With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.

- Some features can be in principle useless and sometimes even noisy. In practice, this means that if the model considers these features when making predictions it could be a problem since its predictions will be influenced by them and the results will be worse.

A good way to solve this issue would be to consider only part of the 1000 features, especially the most relevant ones for the considered task. But how can we establish which features are actually relevant or not for our problem? This cannot be done by hand especially when it comes to applications involving large sets of features.

This can be seen as a combinatorial problem in which we want to find the best subset of features.

A naive approach could be the described by the following algorithm:

Algorithm 1 Naive Feature Selection Algorithm.

```

F ← features {The set of features}
fbest ← ∅ {The best subset of considered features}
ncurr ← ∅ {The current number of considered features}
loss ← defineLossFunction() {Define the loss function to use}
while ncurr ≤ sizeOf(F) do
    fcurr ← getBestSetOfFeatures(ncurr) {Go through all the possible combina-
        tions of ncurr features and get the one producing the best results according
        to the loss function loss}
    if loss(fbest) ≤ loss(fcurr) then
        fbest ← fcurr {Update the best set of features}
    end if
end while
return fbest

```

Obviously this naive approach has a very big issue: this algorithm requires to try an exponential number of different subsets of features and train an exponential number of Machine Learning models! Being more precise, its complexity is $O(k!)$ where k is the number of features.

Training a Machine Learning model is very time consuming. In our case with 1000 features we would have to train $1000 \cdot 999 \cdot 998 \cdot \dots \cdot 2 \cdot 1$ models because of the different subsets of features, that results in $4.0238726007 \cdot 10^{2567}$ models, more than the estimated number of atoms in the entire universe!

At this point you may be wondering that having 1000 features is a very high number, but this is surely not the case. In fact, images usually have thousands or millions of different features: the pixels and metadata.

Anyway, even if we had only 10 features, it would be necessary to train and test 3628800 different models on different subsets of features.

We recall that we cannot perform Feature Selection by hand, since in most of the cases data can have up to thousands or even more features so we need a way to solve this problem.

If we assume that all features are meaningful and not only containing noise, we can make an improvement to this naive approach that consists in observing that the higher the amount of features considered, the more likely it is for the model to perform better. Therefore, we can decide to only consider a fixed number of features k and try to find the best subset of k features without considering all

3.2. FEATURE SELECTION

the subsets of features of size s such that $1 \leq s < k$ and $k < s \leq n$ where n is the total amount of features.

Algorithm 2 Naive approach to Feature Selection improved.

Require: n {The total number of features}

Require: k {The final number of features to keep}

$F \leftarrow \emptyset$ {The set of all the possible features}

$f \leftarrow \emptyset$ {The final subset of features obtained by the algorithm}

$bestLoss \leftarrow \infty$

for all $S \subseteq F$ s.t. $|S| = k$ **do**

$model \leftarrow \text{trainWithFeatures}(S)$

if $\text{loss}(model) \leq bestLoss$ **then**

$f \leftarrow S$

$bestLoss \leftarrow \text{loss}(model)$

end if

end for

return f

Even in this case the number of times we need to train the model according to each possible subset of k features is $\binom{n}{k}$ times, with n the total number of features. This means that even if we are taking into considerations far fewer subsets with respect to the previous Feature Selection naive approach, we still require to train our model an exponential number of times based on the number of features at our disposition.

So, to tackle this problem, we usually make use of some heuristic approaches that allow to reach solutions in a feasible amount of time but with some approximations.

After having seen the problem of Feature Selection arising in the Machine Learning field, we will now consider the Information Retrieval field.

As you can imagine, we can immediately find the Feature Selection problem also here when we require to employ some Learning Models in order to perform any sort of classification task for retrieval purposes. Also when it comes to optimize our systems according to the input collections, it is sometimes necessary to reduce the amount of data to speed-up retrieval or to simply improve effectiveness by avoid considering noisy features.

There are many approaches to solve the problem of Feature Selection. In this case we will consider the Filter methods analyzed in [41].

According to a Filter approach, features will be selected based on information theoretical measures not optimizing the model itself. This means that we employ measures such as variance or entropy to detect which features could bring more information. The model will not be used to understand which features can improve its performance.

As we saw from the previous section, we firstly need to model our problem as a QUBO problem in order to make it solvable by the Quantum Annealer devices. It is possible to consider several QUBO variants. For example, 2 possible strategies could be:

- *MIQUBO*: exploits Mutual Information (MI) to find the best set of features. The matrix Q will be such that each element q_{ij} is defined as

$$q_{ij} = \begin{cases} -MI(f_i, y|f_j) & \text{if } i \neq j \\ -MI(f_i, y) & \text{if } i = j \end{cases} \quad (3.9)$$

where $MI(f_i, y)$ is the Mutual Information between feature f_i and the target y . In this case a penalization term must be introduced since the trivial solution is to have all features selected.

- *QUBO-Correlation*: exploits Pearson's r Correlation to find the best set of features. The matrix Q will be such that each element q_{ij} is defined as

$$q_{ij} = \begin{cases} -r(f_i, f_j) & \text{if } i \neq j \\ r(f_i, y) & \text{if } i = j \end{cases} \quad (3.10)$$

After having formulated the problem in its corresponding QUBO version, another step is necessary in order to solve it through Quantum Annealers: the minor embedding.

The embedding process can be done according to some automated methods, for example the ones provided by the D-Wave libraries, or it can be done manually. In [41], Quantum Annealing approaches have provided almost the same results in terms of effectiveness with respect to classical approaches without the employment of quantum annealers. The advantage that Quantum Annealing provided is mostly in terms of efficiency. In fact, as in [44], it is clear that if the size of the problem fits on the QPU it is possible to retrieve the solutions in almost constant time independently from the number of features that we want to keep.

3.3. OTHER QUANTUM ANNEALING PROBLEMS

If the size of the problem does not fit the QPU, we can still benefit from its usage by means of an Hybrid approach.

This approach consists in using both the quantum annealer itself but also additional classical devices. In particular, classical devices will be used to perform part of the computation and will divide the problem into smaller instances that will fit on the QPU. These instances will be solved and the results will be combined together in order to retrieve the final result for the problem.

This approach provides advantages since it allows to solve a problem having a size that is too large for the QPU. Anyway it has its own drawbacks since involving classical hardware will decrease the overall efficiency by a lot.

3.3 OTHER QUANTUM ANNEALING PROBLEMS

In the section above we saw that Feature Selection can be a challenge solvable by means of quantum annealers. In fact, it is possible to formulate the Feature Selection problem by means of a QUBO model.

Now we want to investigate which other important problems and challenges could be tackled by Quantum Annealing. This can give us a better idea of which Shared Tasks we can devise to start the new QuantumCLEF evaluation campaign.

We want to highlight the fact that quantum annealers can provide the most benefits when performing offline tasks. With offline computation we mean performing some calculations which do not involve any interaction with the final users of the system. In fact, offline computation refers to the process of analyzing and processing large volumes of data to create an index or other data structures that can be quickly searched and retrieved during online search queries.

3.3.1 CLUSTERING

Clustering is a problem in which we want to group together similar objects based on their characteristics or attributes.

Clustering is one of the most common form of Unsupervised Learning problems, which implies that there are no labeled or annotated data. Therefore, the algorithm that we want to implement in order to cluster these items should be able to compute similarities between data according to some predefined distance

metrics [46].

Clustering can be very useful in Information Retrieval to provide alternative search results that are related to a given query, to help users explore a collection of documents, or to organize large collections of documents for easier browsing and faster retrieval times.

We now formalize the clustering problems of k-center, k-means and k-median. First of all we start by the definition of a metric space.

A metric space is an ordered pair (M, d) where M is a set and $d(\cdot)$ is a metric on M , i.e. a function $d : M \times M \rightarrow \mathbb{R}$ such that $\forall x, y, z \in M$ we have that:

$$d(x, y) \geq 0 \quad (3.11a)$$

$$d(x, y) = 0 \text{ if and only if } x = y \quad (3.11b)$$

$$d(x, y) = d(y, x) \quad (3.11c)$$

$$d(x, z) \leq d(x, y) + d(y, z) \quad (3.11d)$$

There are several functions satisfying the above properties such as the Hamming distance, the Jaccard distance and most importantly the Minkowski and Cosine distances.

Now we define P as a set of N points in a metric space (M, d) and we let k be the target number of clusters with $1 \leq k \leq N$. We define a k -clustering of P a tuple $C = (C_1, C_2, \dots, C_k; c_1, c_2, \dots, c_k)$ where

$$P = C_1 \cup C_2 \cup \dots \cup C_k \quad (3.12a)$$

$$c_1, c_2, \dots, c_k \text{ are suitable centers for the clusters and } c_i \in C_i \forall i \in [1, k] \quad (3.12b)$$

So for a given input set P we want to find the best set of k clusters according to a given objective function. Based on the objective function used we have:

- k-center clustering: it minimizes the maximum distance of any point from the center of its cluster.
- k-means clustering: it minimizes the sum of the squared distances of the points from the centers of their respective clusters.
- k-median clustering: it minimizes the sum of the distances of the points from the centers of their respective clusters.

We will now define the objective function of the k-means clustering. Firstly we define $d(x, S) = \min_{y \in S} (d(x, y))$ where $x \in P$ and $S \subseteq P$. Then the objective

3.3. OTHER QUANTUM ANNEALING PROBLEMS

function $\phi_{k\text{-means}}$ of k-means clustering is

$$\phi_{k\text{-means}}(P, S) = \sum_{x \in P} (d(x, S))^2 . \quad (3.13)$$

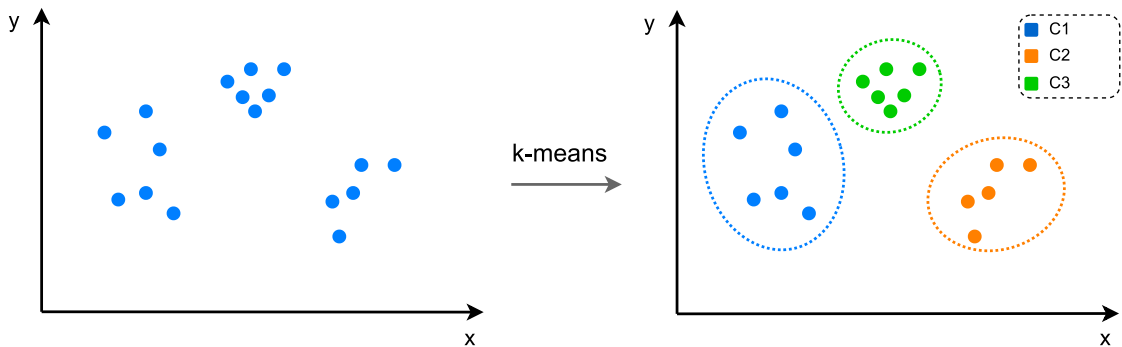


Figure 3.5: Representation of k-means with $k = 3$.

We recall that according to the Vector Space Model, a document can be represented as a vector that lies into a multidimensional space. Therefore, we can apply what we just saw in the case we want to cluster documents or queries.

It is possible to formulate the Clustering problem according to the QUBO formulation [47]. In this way it will be possible to exploit Quantum Annealing to solve the Clustering problem in a lower amount of time and hopefully with better results. This could help improving the performances of IR systems making use of Clustering methods.

In fact, there are already some approximation algorithms that can be employed to solve this problem by means of classical computers, such as the Lloyd's algorithm. Unfortunately, Lloyd's algorithm is an iterative algorithm that can still require an exponential number of iterations in the input size and it does not guarantee to return the optimal solution.

3.3.2 BOOSTING

Boosting is a technique that consists in combining a set of simple (also known as "weak") predictors in such a way as to produce a more powerful (also known as "strong") predictor.

This method can be employed to implement Random Forest models that will be then used for classification purposes.

A Random Forest can be seen as a combination of Decision Trees in such a way

that the final prediction of the Random Forest model will be based on the predictions of each Decision Tree embedded into the model itself.

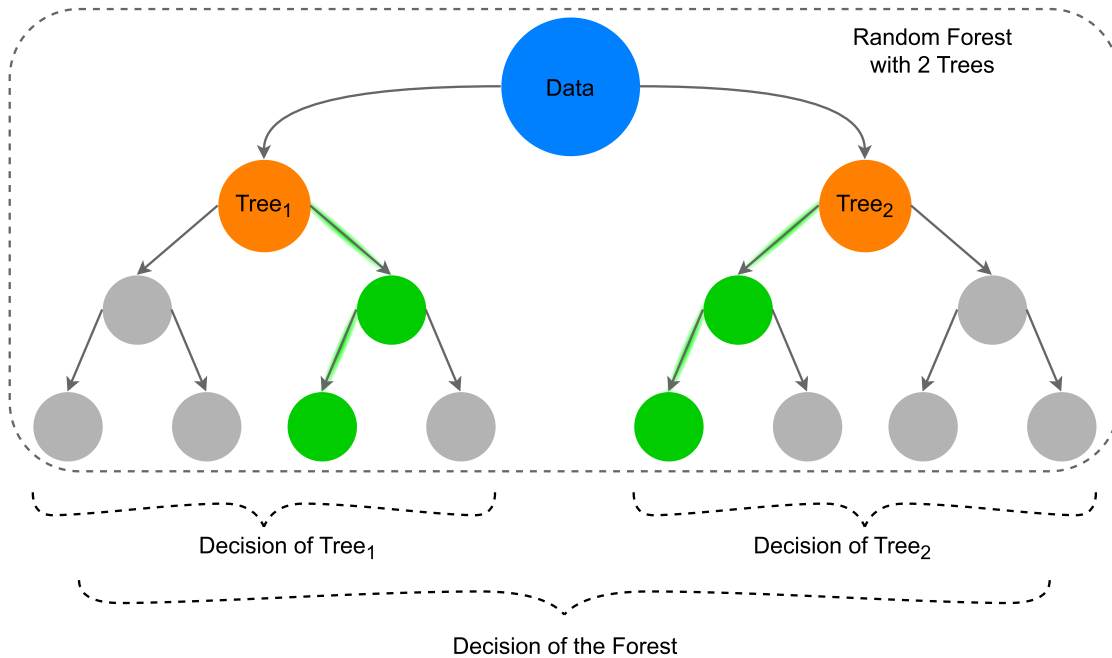


Figure 3.6: Representation of a Random Forest.

A Decision Tree instead is a model that has a hierarchical tree structure consisting of a root node, branches, internal nodes and leaf nodes.

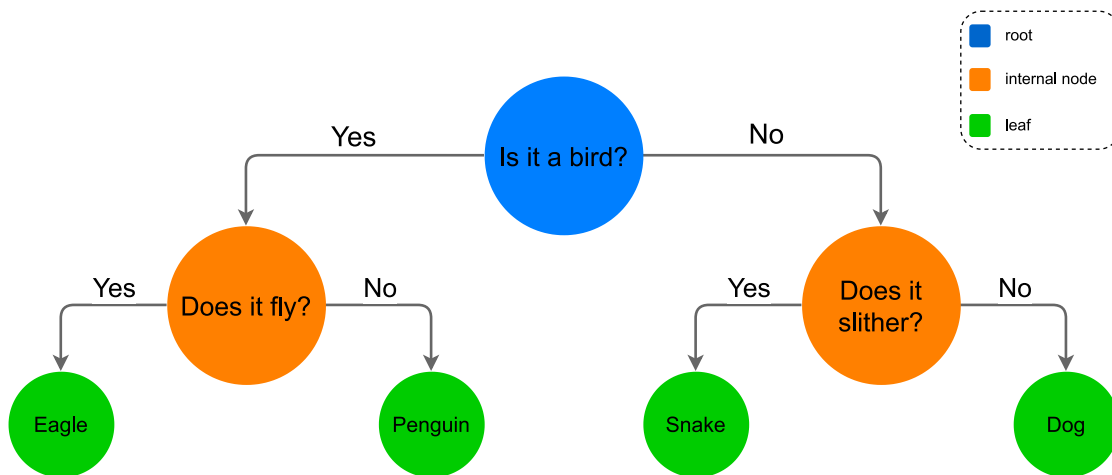


Figure 3.7: Representation of a Decision Tree.

In the example above, we can see that from the data provided as input, we have a sequence of decisions that will bring us to one of the possible leaves that

3.3. OTHER QUANTUM ANNEALING PROBLEMS

correspond to the final prediction for the input data. Each decision is made usually considering one of the features that the input data have. Considering the case above, the input data could be binary vectors of dimension 3 in which we have that:

- 1st component: 1 if the vector represents a bird and 0 otherwise.
- 2nd component: 1 if the vector represents an animal that flies and 0 otherwise.
- 3rd component: 1 if the vector represents an animal that can slither and 0 otherwise.

So if we provide the vector (1, 0, 0) in input to the Decision Tree, we will have that the answer will be "Penguin".

According to the Boosting technique, we can implement a Decision Forest in the following way:

Algorithm 3 Boosting applied to Random Forests.

Require: n {The number of Decision Trees to train}

Require: k {The final number of Decision Trees in the Forest}

$F \leftarrow \emptyset$ {The Random Forest}

$treeSet \leftarrow \emptyset$ {The set of Decision Trees}

$i \leftarrow 0$

for $i \in [0, n]$ **do**

$treeSet.add(\text{trainTree}())$ {Train a new Decision Tree and add it to the set of Decision Trees}

end for

for all $S \subseteq treeSet$ s.t. $|S| = k$ **do**

if $\text{loss}(S) \leq \text{loss}(F)$ **then**

$F \leftarrow S$

end if

end for

return F

In simple words, what is happening here is that we firstly train each Decision Tree and then we try all the possible subsets of Decision Trees of a given size in order to get the one that behaves the best with respect to the considered problem. As you can notice, we have an issue. Considering n the number of Decision Trees

that we have trained and k the final size of the Decision Forest, then we need to enumerate $\binom{n}{k}$ possible subsets which is an exponential number of subsets!

Also in this case, there have been studies [48] [49] [50] showing that it is possible to solve Boosting problems by means of Quantum Annealing. Therefore, it is possible to apply QA devices in order to find the best set of Decision Trees leading to an optimal Random Forests. We want to highlight the fact that Boosting methods can be applied also to other types of classifiers.

An example of Boosting algorithm that is currently applied in the Information Retrieval field is LambdaMART [51], which is an algorithm that uses gradient boosting to optimize Learning to Rank specific cost functions such as NDCG.

4

Shared-Task Proposals

In this chapter we will better formalize the Shared Tasks according to the problems that have been presented in the previous chapter.

In particular, we will propose 3 tasks according to the Feature Selection, Clustering and Boosting problems. These Shared Tasks could be then a starting point for the QuantumCLEF evaluation campaign.

We will also make some considerations regarding the effectiveness and efficiency of quantum annealers. In particular, we will try to establish some common metrics that can be employed in order to account for the uncertainties that happen when solving through Quantum Annealing.

4.1 TASK 1: FEATURE SELECTION

In this task, participants are asked to solve the Feature Selection problem. Even though Feature Selection has already been tackled with the Quantum Annealing paradigm, there can be other improvements concerning the development of the matrix Q in the QUBO formulation. In addition, in order to start the QuantumCLEF evaluation campaign, it is probably fundamental to begin with a baseline that has already been implemented and tested so that the participants will start gaining experience in this field.

Some possible datasets that can be used are the MQ2007 or MQ2008 datasets [52] that has already been employed in a previous work [41]. In this way it is possible to compare the results obtained by the participants with the one obtained previously. In addition, we have the guarantees that the dataset will have

4.2. TASK 2: CLUSTERING

a small enough number of features, therefore it will be possible to deploy the problem into the QPU without many issues.

A possible variation of the task could be to set a fixed number of features which is much smaller than the actual number of available features and let the participants find which subset of features is the one that allows their systems to perform the best.

In this case, to evaluate the effectiveness of the Quantum Annealing approaches developed by the participants we could employ some retrieval algorithms that will employ the subset of selected features to retrieve the documents corresponding to some given queries. Then it will be possible to assess the effectiveness of these methods according to the list of retrieved documents by means of evaluation measures such as the Normalized Discounted Cumulative Gain.

4.2 TASK 2: CLUSTERING

In this task, participants are asked to solve the Clustering problem.

In this case the dimensionality of our data (namely the number of features) do not impact the resolution of the problem considering the QPU. In fact, the QUBO formulation should be such that each component of the matrix Q is related to the distance among the considered datapoints. It could be possible to use a dataset in which we have audio features (e.g., time domain audio features such as Mean Absolute Value, Variance...) and we need to group the audio tracks according to their similarity. In this case we could consider using the audioMNIST dataset [53] that contains 30000 audio samples in which humans have pronounced the 9 digits. In this case labels are available so we can measure the performance of the Clustering algorithm proposed by the participants against the given labels. Participants are invited to employ techniques to extract time-domain features from the samples to use them when Clustering.

It is also possible to apply this to a set of documents where each document will be interpreted as a vector as in the Vector Space Model. The dataset in this case could be the BBC dataset [54] containing 2225 different documents belonging to 5 categories. Also in this case labels are available, therefore we can measure the quality of the clusters according to the categories themselves.

In order to make future participants more familiar with the Clustering problem and with the D-Wave Python libraries, I provide here a simple script that solves the Clustering problem for a simple dataset of points and also gives the possi-

bility to interact with the D-Wave inspector tool. The D-Wave inspector makes it possible to visualize what is happening inside the QPU.

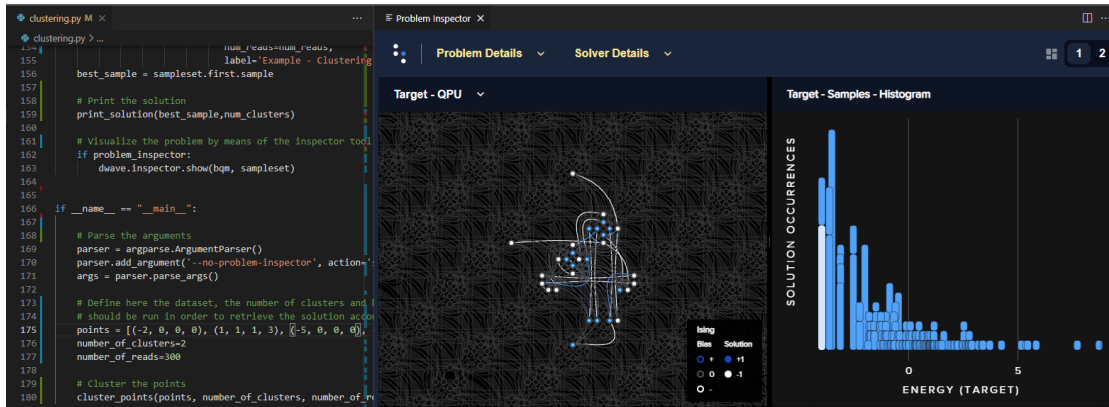


Figure 4.1: The graphical interface with the D-Wave inspector tool that shows how the problem has been embedded in the QPU.

```

1 import math
2 import numpy as np
3 import dwavebinarycsp
4 import dwave.inspector
5 import argparse
6 from dwave.system import EmbeddingComposite, DWaveSampler
7
8
9 class Point_Cluster:
10     """
11     Class representing a point and all the possible clusters it can
12     be associated to.
13     Each cluster is represented by an integer number.
14     Given a point x and a number of clusters k, the point can be
15     associated to all k clusters in
16     principle, so we need to take into accounts all the possible
17     associations by means of
18     binary variables when constructing the problem
19     """
20     def __init__(self, coordinates, num_clusters):
21         """
22         Args:
23         coordinates (Tuple):
24             The coordinates of the point
25         num_clusters (int):
26             The number of clusters
27         """
28         self.coordinates = coordinates
29         self.clusters=[]

```

4.2. TASK 2: CLUSTERING

```
27         for i in range(num_clusters):
28             self.clusters.append("(coordinate={}, cluster={})".format
29                 (str(coordinates),i))
30
31
32 def get_distance(point_1, point_2):
33     """Calculates the euclidean distance between 2 points
34
35     Args:
36         point_1 (Point_Cluster):
37             The first considered point
38         point_2 (Point_Cluster):
39             The second considered point
40     """
41     return math.dist(point_1.coordinates, point_2.coordinates)
42
43
44 def get_max_distance(points):
45     """Calculates the maximum distance between all the couples of
46     points
47
48     Args:
49         points (list of Point_Cluster):
50             The list of all the points to be clustered
51     """
52     max_distance = 0
53     for i, point_1 in enumerate(points[:-1]):
54         for point_2 in points[i+1:]:
55             distance = get_distance(point_1, point_2)
56             max_distance = max(max_distance, distance)
57
58     return max_distance
59
60 def print_solution(solution, num_clusters):
61     """Prints the solution in a more human-readable way
62
63     Args:
64         solution (Set):
65             The samples representing the solution to the problem
66         num_clusters (int):
67             The number of clusters
68     """
69     clusters={}
70     for i in range(num_clusters):
71         clusters[i]=[]
72
73     for point, associated in solution.items():
```

```

74     if(associated==1):
75         result=str(point)[1:len(str(point))-1]
76
77         cluster=int(result.rsplit('=', 1)[1])
78
79         point=(result.rsplit('=', 2)[1])
80         point=(point.rsplit(',', 1)[0])
81         point=eval(point[1:len(point)-1])
82
83         clusters[cluster].append(point)
84
85     for cluster, points in clusters.items():
86         print("Cluster {} has associated points {}".format(cluster,
87 points))
88
89 def cluster_points(points, num_clusters, num_reads, problem_inspector
90 ):
91     """Perform clustering analysis on given points
92
93     Args:
94         points (list of tuples):
95             The Points to be clustered
96         num_clusters (int):
97             The number of clusters
98         num_reads (int):
99             The number of times the problem is run to obtain good
100 statistics
101         problem_inspector (bool):
102             Whether to show problem inspector
103     """
104     # Set up problem
105     # Note: max_distance gets used in division later on. Hence, the
106     max(.., 1)
107     # is used to prevent a division by zero
108     point_clusters = [Point_Cluster(point, num_clusters) for point in
109 points]
110     max_distance = max(get_max_distance(point_clusters), 1)
111
112     # Build constraints
113     csp = dwavebinarycsp.ConstraintSatisfactionProblem(dwavebinarycsp
114 .BINARY)
115
116     # Apply constraint: each point can only be in one cluster
117     choose_one_group = set()
118     for i in range(num_clusters):
119         constraint=np.zeros(num_clusters)
120         constraint[i]=1
121         choose_one_group.add(tuple(constraint))

```

4.2. TASK 2: CLUSTERING

```
117
118
119     for point in point_clusters:
120         csp.add_constraint(choose_one_group, tuple(point.clusters))
121
122     # Build initial BQM
123     bqm = dwavebinarycsp.stitch(csp)
124
125     # Edit BQM to bias for close together points to share the same
126     color
127     for i, point1 in enumerate(point_clusters[:-1]):
128         for point2 in point_clusters[i+1:]:
129             # Set up weight
130             d = get_distance(point1, point2) / max_distance #
131             rescale distance
132             weight=d
133
134             # Apply weights to BQM
135             for k in range(num_clusters):
136                 bqm.add_interaction(point1.clusters[k], point2.
137                 clusters[k], weight)
138
139     # Edit BQM to bias for far away points to have different colors
140     for i, point1 in enumerate(point_clusters[:-1]):
141         for point2 in point_clusters[i+1:]:
142             # Set up weight
143             d = get_distance(point1, point2) / max_distance
144             weight=-d
145
146             # Apply weights to BQM
147             for c1 in range(num_clusters):
148                 for c2 in range(num_clusters):
149                     if(c1!=c2):
150                         bqm.add_interaction(point1.clusters[c1],
151                         point2.clusters[c2], weight)
152
153     # Submit the problem to the D-Wave sampler
154     sampler = EmbeddingComposite(DWaveSampler())
155     sampleset = sampler.sample(bqm,
156                               chain_strength=4,
157                               num_reads=num_reads,
158                               label='Example - Clustering')
159     best_sample = sampleset.first.sample
160
161     # Print the solution
162     print_solution(best_sample, num_clusters)
163
164     # Visualize the problem by means of the inspector tool
```

```

162     if problem_inspector:
163         dwave.inspector.show(bqm, sampleset)
164
165
166 if __name__ == "__main__":
167
168     # Parse the arguments
169     parser = argparse.ArgumentParser()
170     parser.add_argument('--no-problem-inspector', action='store_false',
171                         dest='problem_inspector', help='do not show problem inspector')
172     args = parser.parse_args()
173
174     # Define here the dataset, the number of clusters and how many
175     # times the problem
176     # should be run in order to retrieve the solution according to
177     # statistics
178     points = [(-5, 0, 0, 0), (1, 1, 1, 1), (2, 4, 4, 4), (3, 2, 2, 2),
179              (5, 2, 2, 2)]
180     number_of_clusters=2
181     number_of_reads=300
182
183     # Cluster the points
184     cluster_points(points, number_of_clusters, number_of_reads, args.
185                   problem_inspector)

```

Code 4.1: Code snippet related to the Clustering problem.

The Clustering problem might require to cluster several datapoints, therefore the size of the problem might not fit on the QPU. A possible way to solve this issue is to iteratively assign part of the total amount of points to clusters and then produce an overall solution that considers all the solutions produced in the iterative process. In fact it is possible to keep the centroids obtained for each iteration in memory. Each centroid will be associated with a weight that represents how many samples were associated to cluster it belongs to at that iteration.

After having performed the clustering considering all the points in the dataset, then we repeat the algorithm once again using the centroids that we have found so far as datapoints with their weights.

This should be repeated until we have the final centroids and at that point we know the points associated to them.

We highlight the fact that this procedure might require much time since it requires several problems to be embedded on the QPU and this requires much time.

We also highlight that through this approach that is a sort of Weighted K-means

4.2. TASK 2: CLUSTERING

approach we are actually employing an approximation algorithm, therefore the final solution can be different from the optimal one considered the given distance function.

Algorithm 4 Approximation of the Clustering problem.

Require: k {The number of clusters}

$P \leftarrow points$ {The set of points to cluster}

$p_i \leftarrow P$ {The set of centroids with their weights at iteration i }

while p_i not fits on the QPU **do**

$S = s_1, s_2, \dots, s_j \leftarrow partition(p_i)$ {Partition p_i into subsets such that each s_1, \dots, s_j fits on the QPU}

for all $s \in S$ **do**

$centroids, weights \leftarrow clusterQA(s, k)$ {Cluster the set of points through the quantum annealer. To each centroid we associate a weight that corresponds to the number of points associated to the considered cluster.}

$p_{i+1}.add(centroids, weights)$ {Update the set of points with the found centroids and weights. }

end for

end while

$centroids, weights \leftarrow cluster(p_i, k)$ {Cluster the final set of points}

$solution \leftarrow \emptyset$

for all $point \in P$ **do**

$cluster \leftarrow getClosestCentroid(p_i.centroids, point)$

$solution.add(cluster, point)$

end for

return $solution$

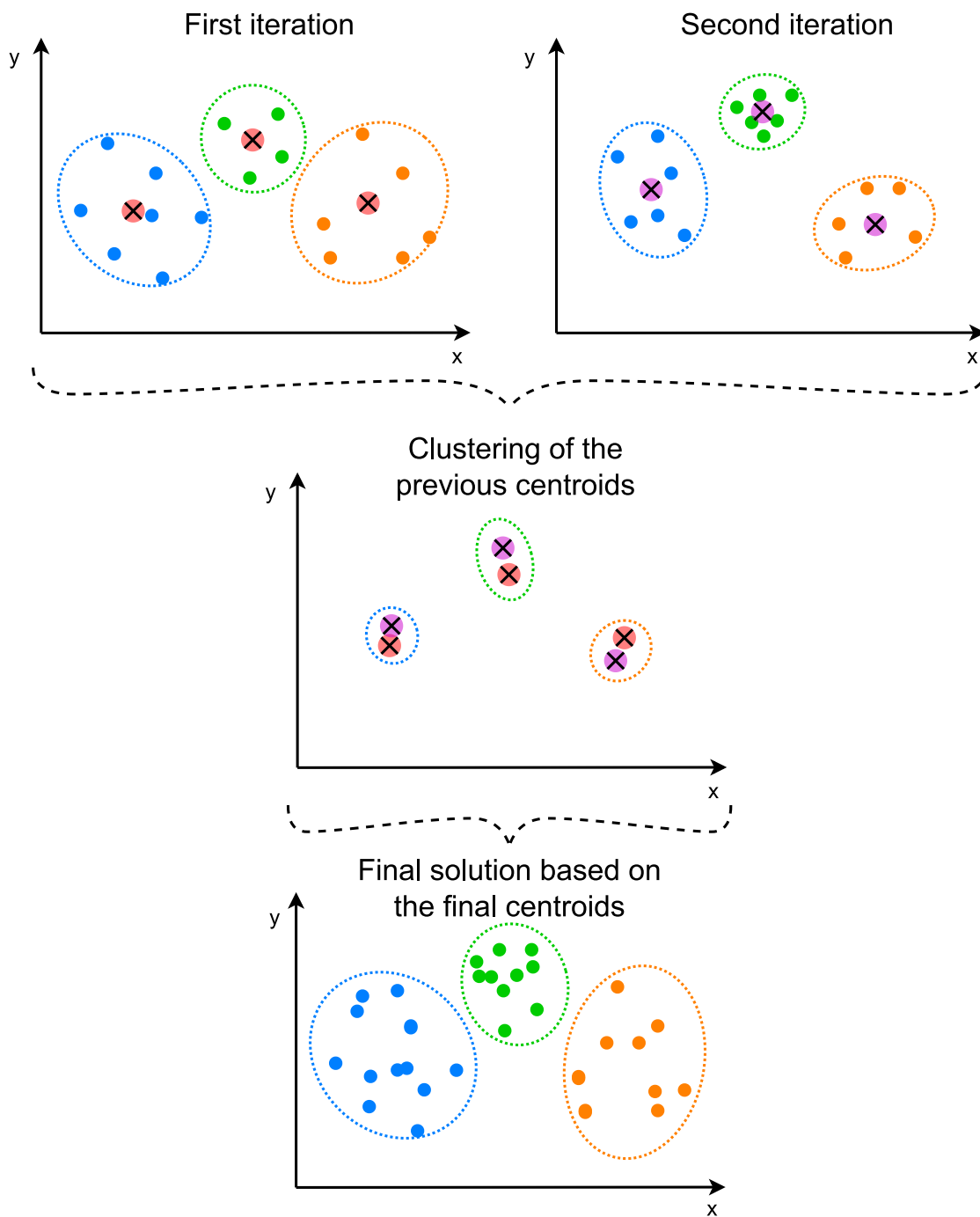


Figure 4.2: Example of the Iterative Clustering approach. This can be used to solve the issue of fitting the Clustering problem on the QPU that has a limited number of qubits and connections available.

4.3 TASK 3: BOOSTING

In this task, participants are asked to solve the Boosting problem. As we have already seen, Boosting is a technique that can be applied to obtain a strong classifier from several weak classifiers by combining them appropriately.

In this case, participants are given a Binary Classification Problem on a small dataset. They need to train several weak classifiers (e.g., Decision Trees having a small amount of levels) and understand how to combine a given number of them appropriately.

In this case, the Training phase of the classifiers will likely be carried out on a classical hardware while the final subset of classifiers will be chosen by means of the quantum annealer. In fact, participants are asked to find the best subset of k classifiers $C_{opt} \subset C$ where C is set of n trained classifiers, such that

$$C_{opt} = \arg \min_{C' \subset C: |C'|=k} \text{loss}(C') \quad (4.1)$$

In this case, to make everything easier for the participants, we can make use of a simple dataset such as the Optical Recognition of Handwritten Digits Data Set [55]. This dataset is provided in the *scikit-learn* [56] Python library.

This task as an IR task can be seen as an image classification task that can be used to correctly label the images so that when users are browsing images by some specific keywords the corresponding images will be returned. This is actually particularly important because providing labels and metadata to images is fundamental to retrieve them correctly in a small amount of time.

To evaluate the effectiveness of the participants' approaches we can apply as evaluation measure the Precision.

4.4 ADDITIONAL SUB-TASKS

Other than these tasks, there could be some sort of sub-tasks involving researchers from the Operations Research and Mathematical field.

For example, due to the hardware limitations of the QPU used, sometimes it is impossible to perform the embedding of the optimization model found.

To overcome this issue, Hybrid approaches are used. We recall that Hybrid approaches make use of classical computers in order to pre-process the problem

and divide it into sub-problems that can be embedded into the QPU in an automatic way.

It might be possible to find some convenient ways to partition the problem into sub-problems according to the nature of the problem itself. Therefore, it could be very interesting to study some specific divide-and-conquer approaches to divide the problem into smaller instances and then combining together the solutions to find the overall solution of the initial problem.

To provide an example, a possible approach that can be used to solve Clustering problems even when they do not fit on the QPU has been discussed above.

4.5 UNCERTAINTIES OF QUANTUM ANNEALING

It is important to highlight that when submitting a problem to a quantum annealer, it is possible to specify the number of solutions to sample for the given problem.

In fact, the solutions for a given problem can vary because of several reasons starting from the physics itself and possible noise and errors introduced.

Usually when submitting a problem, hundreds or thousands of samples are taken into consideration to produce the final result, which will be the best solution among the ones produced by the quantum annealer.

In this nondeterministic scenario, it can be useful to compare different submissions considering all the solutions produced by the quantum annealer for a given problem. In this way, it will be possible to compare different submissions also considering statistical measures such as the Analysis of Variance (ANOVA) measure that considers the variance of the obtained to compare different submissions.

This approach will allow us to additionally produce some general statistics regarding the average quality of the solutions provided by Quantum Annealing technologies.

4.6 EFFICIENCY OF QUANTUM ANNEALING

Estimating the efficiency of the quantum annealers with respect to classical computers is complex. In fact, it is necessary to keep into consideration several factors impacting on the total amount of time required to solve the considered

4.6. EFFICIENCY OF QUANTUM ANNEALING

problem.

First of all, we need to consider the time required in order to formulate the problem as a QUBO problem. This procedure does not come for free and it can take some time if the problem has a large size.

Once we want to solve a problem it is necessary to submit it to the D-Wave quantum annealers. This is done by sending the problem through the network to an endpoint machine that will then embed the problem inside the quantum annealer itself. As you can imagine, network latencies are introduced in this step. The problem is that network latencies are difficult to estimate with an high grade of precision because of their variability. In fact, it might happen that due to a temporary network congestion the problem is received even after some seconds.

After the problem has been received, it needs to be embedded into the QPU of the quantum annealer chosen. This can be another time-demanding step especially if the problem must undergo an hybrid approach due to its high size. We will refer to the embedding time as T_e in this section. We highlight also that usually embedding is done by means of probabilistic algorithms and therefore it is likely that the problem will be embedded in different ways if it is executed more times.

After the embedding phase, the problem is ready to be solved by the quantum annealer. In this case it is possible to divide the amount of time employed to solve the problem instance into several times to account for specific action taken by the quantum annealer itself.

Firstly the annealer needs to set some internal parameters and perform some low level operations in order to solve the problem. This is done in the initial phase and it takes an amount of time called *Programming Time* T_p .

After this phase, the actual sampling occurs. Sampling is the phase in which the actual annealing happens. It is usually done several times in order to retrieve the best solution according to the given problem (i.e. hundreds or thousands of times). This *Sampling Time* T_s can be broken down into smaller pieces as follows:

- *Anneal Time per sample*: the time that the actual annealing phase takes. This time is usually specified by the developer itself. It is usually set to 20 μ s.
- *Readout Time per sample*: the time employed to read the values of the qubit at the end of the annealing process. We recall that this is done only at the end to let the system evolve naturally during its annealing phase.

- *Delay Time per sample*: the time that is required to bring the QPU to its initial state in order to repeat the experiment.

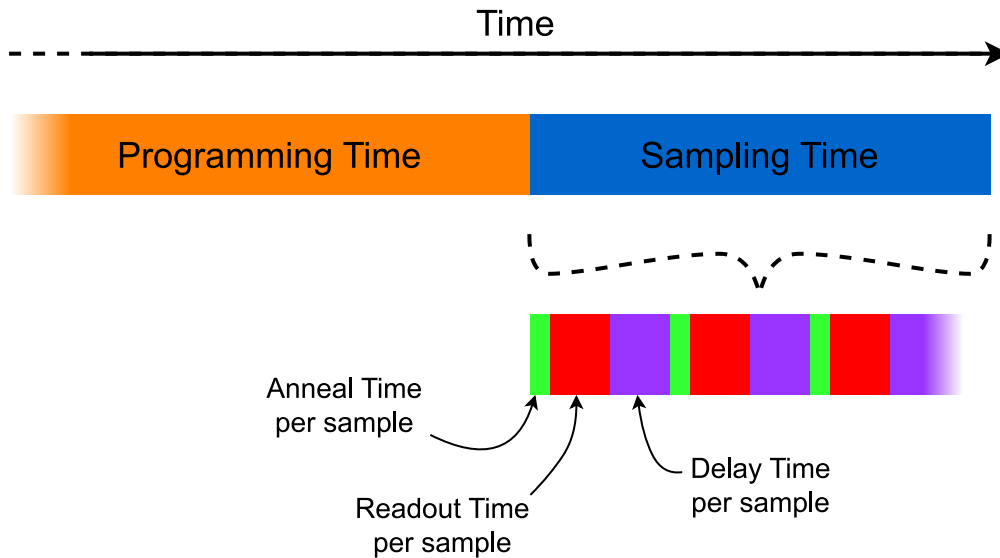


Figure 4.3: Representation of the various QPU times corresponding to the different phases when solving a problem through the D-Wave quantum annealer.

In order to be "fair" when comparing the time required by the quantum annealers with respect to classical machines when dealing with some problems, it is necessary to avoid considering the network latencies.

To provide different levels of comparison, it could be very interesting to compare the time required by the 2 different architectures by considering the following cases:

- $T_p + T_s$ vs classical hardware: this measures the actual amount of time required for the annealing phase only compared to a classical machine.
- $T_e + T_p + T_s$ vs classical hardware: this measures the time required for the embedding and the annealing phase together compared to a classical machine. We can notice that the embedding time can be high and can influence the efficiency a lot when dealing with small problem instances. On the other hand, when problem instances grow the embedding time does not grow as fast as the time required by the classical machine to solve them, therefore we expect the quantum computers to be able to solve big instances in much less time with respect to classical machines.

We want to highlight that the Anneal Time, Readout Time and Delay Time are almost constant and in the range of some microseconds.

To perform a fair comparison, participants are invited to solve the tasks according

4.6. EFFICIENCY OF QUANTUM ANNEALING

to a classical approach and a Quantum Annealing approach using the machines we will provide them in our Submission System that will be described in the next chapters.

5

Design of the Infrastructure

In this chapter we will focus on the design of the Submission System in its entirety, starting from a general overview of all the necessary components of our system.

Furthermore, we will provide a brief introduction about Kubernetes and some of its most important objects. Kubernetes will be then used to deploy our Submission System and the implementation will be described more in depth in the next chapter.

Finally we will discuss about the Web Application by designing the database and addressing all the most important security vulnerabilities that we must mitigate to ensure that our data will be protected both from theft and damage. Cybersecurity is in fact a non-negligible part that must be considered when building a Web Application that requires to deal with data provided by the users.

5.1 THE SUBMISSION SYSTEM

To allow users to carry out the tasks it is required to design a system that will be simple, secure and scalable.

In fact, it is not known the exact number of participants that will take part into this evaluation campaign and it is likely that new editions of the QuantumCLEF evaluation campaign will have more involved participants because the Quantum Annealing field is likely to attract more attention in the future, especially when

5.1. THE SUBMISSION SYSTEM

quantum annealers will be furtherly improved.

We provide here below a high-level diagram representing how the system is designed.

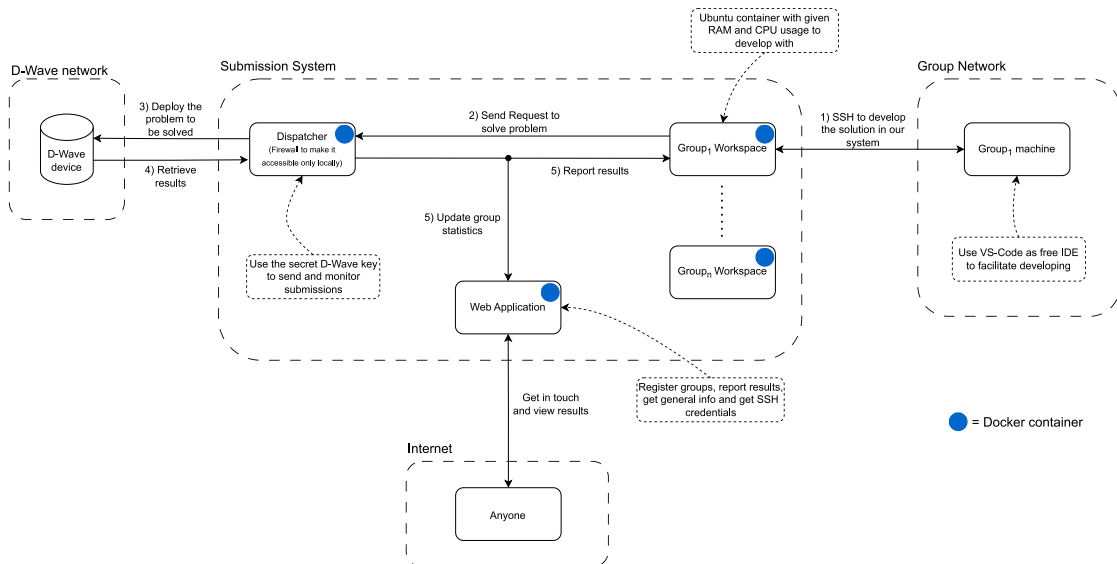


Figure 5.1: Representation of the design of the submission system. Docker containers are being used in order to make everything more scalable and secure.

As you can see, the Submission System is composed of several parts:

- A Dispatcher: this is a component that is in charge of forwarding the requests of using the quantum annealer to the D-Wave quantum annealer and of providing the corresponding results back. This is done because in this way it is possible to monitor the access to the quantum annealer by each group ensuring that groups do not exceed their given quotas. In addition, to access the quantum annealer it is required to have a Secret Key that we cannot give to our users. This key will be known only by the administrators and will be used by the dispatcher to communicate with the quantum annealer. As you can imagine, the dispatcher needs to be protected in such a way that only machines internal with respect to the Submission System can have access to it. In this way we will keep the Secret Key safe from a possible malicious usage.
- A Web Application: this component will be used to provide general information to external users about the tasks. In addition, through the Web Application, each group participating into one of the tasks can monitor its quotas by means of a simple dashboard. The Web Application will be used by the organizers and administrators to register groups participating into the tasks and provide information about the campaign and the results obtained so far.

To do so, it is required that our Web Application has a login system to access the protected areas.

The Web Application will make use of a Relational Database to save the required data such as accounts, current and past Shared Tasks, submissions by each group etc...

- Personal groups' containers: these containers will be accessed through the secure SSH protocol in order to let the group develop and test their solutions in our environment. This is done in order to avoid having users to download our libraries and packages in their own machines and for reproducibility reasons.

Each group will have its own credentials in order to access its corresponding container. Since accessing through a terminal may not be so user friendly, we suggest groups to use the free Visual Studio Code Integrated Development Environment (IDE) which allows to connect through the SSH protocol to a host by means of a simple interface.

Each container will also contain a preconfigured git repository so that each group will share its solution with the internet. In such manner it will be possible to share the knowledge and to repeat the experimental results that groups have obtained.

To better understand how the communication happens in the provided Submission System, we provide here a diagram representing the communication between the various components presented:

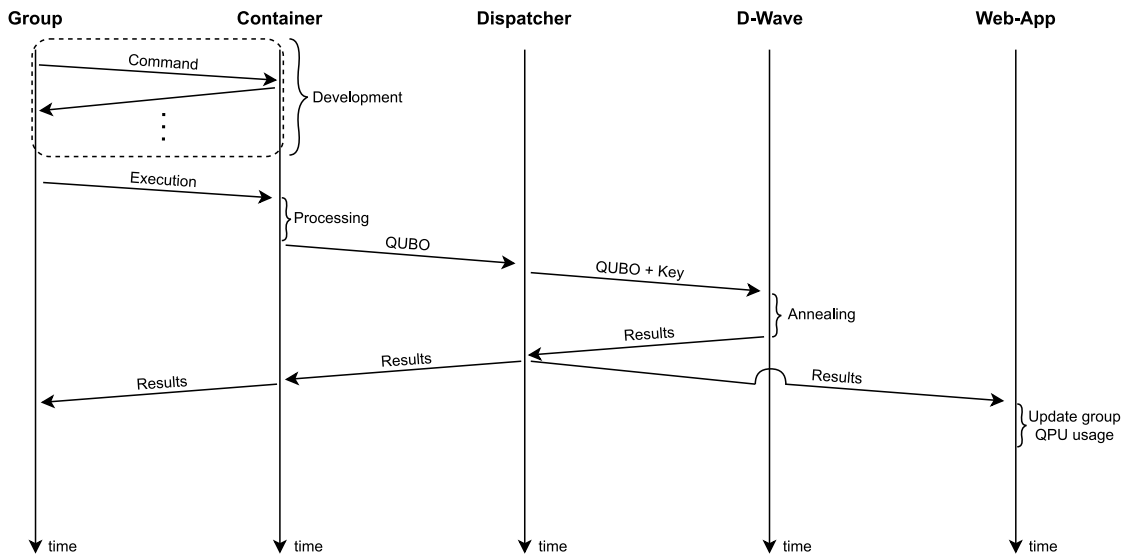


Figure 5.2: Representation of how the communication takes place in the Submission System starting from the personal machine of a group to the quantum annealer.

To ensure that the system is scalable, we need to build a system that is likely composed of several machines (or nodes). In addition, we must ensure that if

5.2. OUR SUBMISSION SYSTEM AS A DISTRIBUTED SYSTEM

we add additional machines we are not required to design and implement the entire system from scratch once again.

To avoid having too many machines involved in our system, we decided to apply virtualization software to run the various components in isolated environments. In particular, we decided to make use of Containers since they allow a more effective resource usage and also higher scalability thanks to the orchestration platforms that are available nowadays.

5.2 OUR SUBMISSION SYSTEM AS A DISTRIBUTED SYSTEM

The system is made of several components that need to cooperate in order to work correctly. We can refer to our entire Submission System as a Distributed System (DS), which can be defined as a collection of independent components usually located on different machines that share messages with each other with the aim of achieving a common goal.

We want to highlight that the components do not need to be situated on different machines in order to have a Distributed System. It is indeed possible to have isolated components that are hosted on the same machine such as containers or virtual machines. The only difference relies on the communication protocols employed. In fact if all the containers or virtual machines are located on a single host machine, it is possible to employ protocols utilizing shared-memory areas in order to exchange data.

A Virtual Machine (VM) is a sort of emulator of a computer system. In other words, it is possible to simulate the execution of one or more isolated computer systems (the virtual machines) on a single host machine as if they were physical machines. Virtual machines have been developed and employed for many years and are still used nowadays in several fields.

On the other hand, containers are a viable and more lightweight alternative to VMs [57]. In fact they can be more efficient because they do not require the overhead of emulating a complete hardware environment as it happens instead for virtual machines. In simple words, containers provide virtualization at the Operating System (OS) level while VMs at the hardware level.

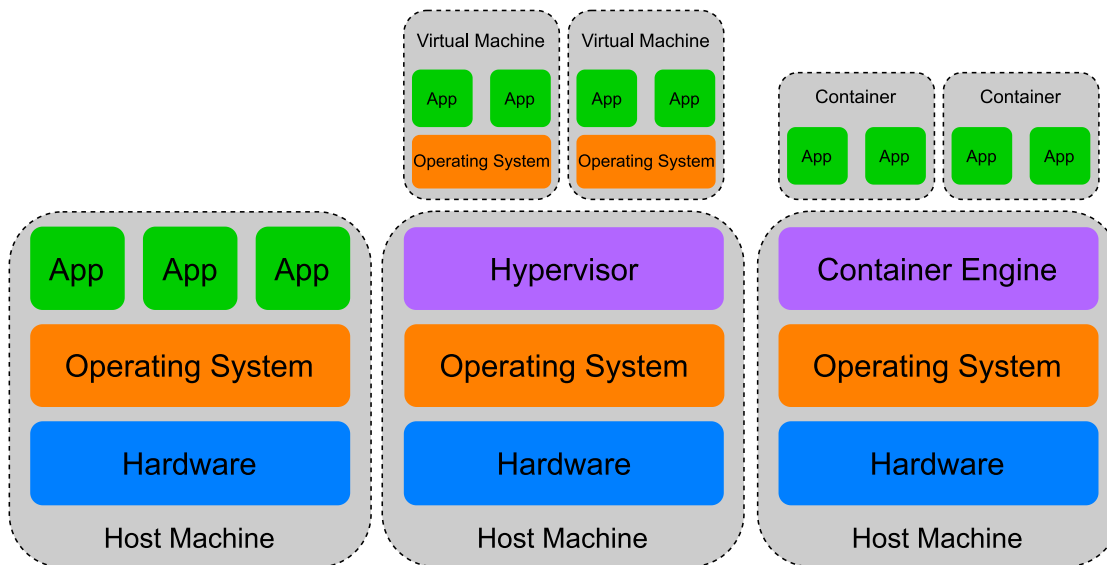


Figure 5.3: An example of the difference between the deployment of some applications on a host machine, on virtual machines and on containers.

According to the arguments that we presented above, we decided to employ containers rather than virtual machines for the actual implementation of our Submission System.

When developing a Distributed System, we usually require several components working together and we aim to build a system that is:

- **Scalable:** the system should be able to increase or decrease the resources employed in order to satisfy all the users that are interacting with the system at a given time.
We can consider as an example a Web Application. The Web Application can be implemented by means of a Distributed System that deploys its several components (e.g. the server, the database) across different machines that are called nodes. In this way, we are *scaling horizontally* the Web Application because we are employing several machines that will serve a portion of clients dealing with our Web Application. As you can imagine, thanks to this approach it is possible to have a much higher number of users with respect to the situation in which we have a single machine that has to deal with all the users.
Furthermore, if the Distributed System is well implemented we can increase the number of nodes to improve the scalability even more without having to implement the system from scratch.
- **Secure:** the system should not expose important and private data to the external users. In fact, the system should use some specific techniques in order to mitigate possible attacks by malicious users that try to steal our data.
For example, a Distributed System should provide firewalls to limit the

5.3. CONTAINER ORCHESTRATION: KUBERNETES

access to specific nodes and services and should employ encryption techniques in order to make the data not understandable by users that are not expected to access them.

- Fault tolerant: the system should be able to handle failures in the correct way.

Failures must be expected in a Distributed System therefore the system should be able to recover from a failure without the corruption of its state. Fault tolerance can be achieved for example by adding replicas of a component so that even if it fails, there will be another component able to substitute it immediately.

We can consider the following example to make things more clear. Consider the case in which we have 20 machines working and each machine is a fundamental component of our Distributed System. From our analysis each machine has a probability of failing that is $P_{failure}(machine) = 1\%$. If we do not have replicas and the system requires all the machines to be constantly working in order to satisfy the user needs, then the probability of the system failing can be seen as the probability of having at least a machine that fails, which is $P_{failure}(system) = (1 - \frac{1}{100})^{20} \approx 18.2\%$.

As you can see the probability is high and it gets higher if we have more and more machines, therefore we need to expect failures and act upon them accordingly.

5.3 CONTAINER ORCHESTRATION: KUBERNETES

To develop our Submission System by means of containers it is required to create a communication network that will be used to exchange messages from one container to the other. The network is mandatory since we want our containers to be possibly located on different machines for scalability purposes.

Managing and handling a high number of containers can be very difficult, especially when it comes to dealing with failures. For this purposes, container orchestration tools are being employed.

Container orchestration can be used to automate the deployment, management, scaling and networking of containers. There are several container orchestrators available nowadays such as Kubernetes, Docker Compose and many more.

In our Submission System we decided to make use of Kubernetes, which is an open-source container orchestrator provided by Google. It has been developed for several years and many cloud services are offering cluster-management solutions based on Kubernetes because of its high level of robustness and reliability. To deploy a distributed application using containers by means of Kubernetes it

is necessary to define and use some objects. Objects are entities in a Kubernetes cluster that are used to manage the cluster's state. This means that we can create high level objects that will be used to represent the cluster's desired state. Kubernetes will then manage autonomously the state of the cluster based on the provided objects.

Some fundamental objects are the following ones:

- **Pod:** an object which is the smallest unit of deployment in Kubernetes. Each Pod is situated in a node and has its own IP address that will be used to exchange data. Each Pod usually hosts a single container but can host more containers if needed. Kubernetes allows to specify the CPU and RAM limits for each Pod through specific properties. In particular, it is possible to specify how many CPUs are reserved to a POD, the fraction of the CPU usage and the maximum amount of RAM memory that can be employed.
- **Deployment:** an object that is used to tell Kubernetes how to manage the pods related to a given application. A Deployment can be used to scale the number of replica pods, enable the rollout or roll back to different application versions if necessary.
- **Service:** an object that is used to abstract and expose an application that is running by means of some Pods. By means of a Service we can represent a set of pods having the same functionality and set the policy for accessing those pods. Services provide an abstracted Service name and IP address to communicate with the considered pods. In addition, services also provide discovery and routing functionalities between the pods.
- **PersistentVolume and PersistentVolumeClaim:** two objects used in order to store permanent data. On-disk files present in a container are ephemeral, which means that whenever the container is stopped all the files will be lost. This presents some problems for some applications when running in containers, especially for databases. A PersistentVolume is a storage resource located in the cluster itself while a PersistentVolumeClaim is a request for a storage resource. A PersistentVolumeClaim is a declaration of need for a storage that will be satisfied according to an actual PersistentVolume. Substantially, a PersistentVolumeClaim is an additional layer of abstraction in which we do not explicitly select which PersistentVolume to choose but Kubernetes will pick an appropriate PersistentVolume to meet that claim.
- **Secret:** an object that contains a small amount of confidential data such as passwords, tokens or keys. By using secrets there is no need to include the given confidential data in your application code thus making the application itself more secure.

We only reported some of the objects that can be found in Kubernetes, in particular the ones that are used the most.

5.4. WEB APPLICATION

To create a Kubernetes object it is possible to issue specific commands with the command line tool but a better way is to create it by writing specific *yaml* files that will be used to represent the object that we want to create. In this way it is possible to edit the files in the future if we simply want to make some adjustments to the cluster.

Kubernetes also provides another very useful feature, which is that if we modify and provide to Kubernetes one or more of the aforementioned files, Kubernetes will automatically perform the changes adding some additional meta-data that allow us to keep track of the versions of our system.

5.4 WEB APPLICATION

In this section we are going to make a discussion about how the Web Application should be designed, starting from the database.

We will also address some potential security issues that should be taken into consideration when developing a Web Application.

5.4.1 WEB APPLICATION: DATABASE

As already discussed, in order to keep track of what has been done in the evaluation campaign it is required to have a database to make our application preserve a state, thus being stateful.

In this case we consider to employ a relational database to store our data, which is a common practice when developing and implementing a Web Application .

The requirements are the following:

- Store the accounts of groups and administrators/organizers.
- Allow groups to participate in a task only after the submission of an application. The submitted applications will be then analyzed by the administrators who will accept or reject them.
The rejection of an application can be due to the fact that there are too many participants and it is necessary to admit only some of them to the tasks.
- Keep track of the submissions of each group in order to estimate the QPU usages and obtain some statistics to fine-tune the resource needs for future QuantumCLEF editions.
In addition, keep also track of the submitted QUBOs in order to understand the evolution of the solutions submitted by the participating groups.

To ensure that all the requirements are satisfied, we started by designing an Entity-Relationship (ER) schema that represents at high level how the database should be structured. This is a fundamental step when developing a Database. In fact, this allows to better understand the constraints and requirements that must be satisfied.

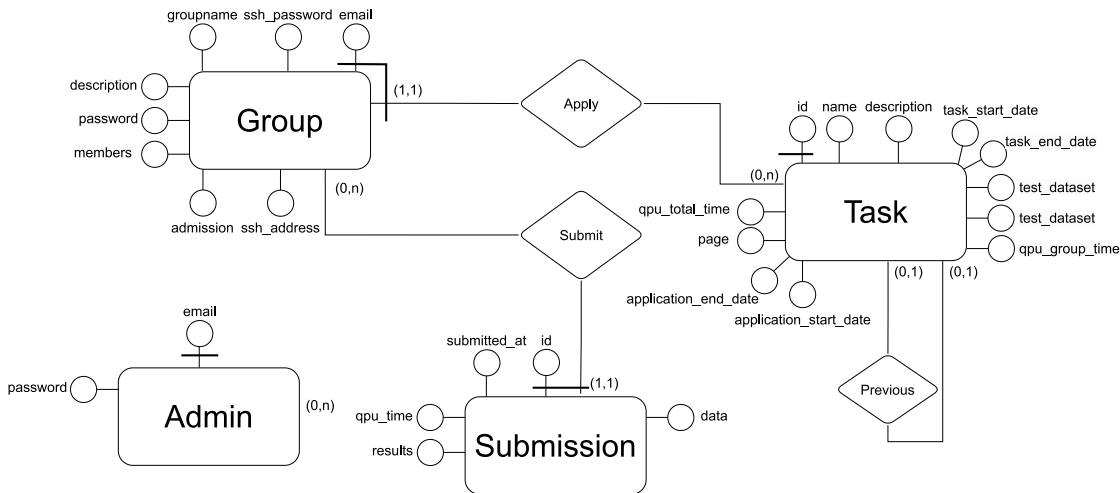


Figure 5.4: Entity-Relationship schema of the SQL database used to manage the Web Application.

From the above ER schema it is possible to derive the corresponding Logical schema below that better represents how the final database structure will look like.

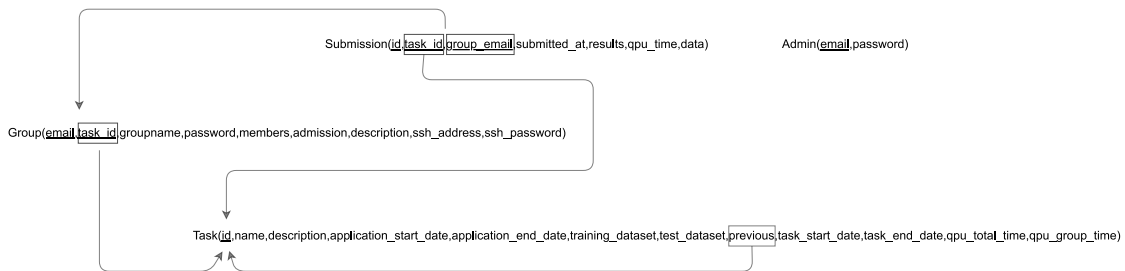


Figure 5.5: Logical schema of the SQL database used to manage the Web Application.

5.4.2 WEB APPLICATION: BACK END AND FRONT END

The Web Application requires to communicate with the database to provide the content to the Web. Since the structured data stored in a database is just a bunch of tuples, we need to transform it in a more visual appealing way.

5.4. WEB APPLICATION

To do so, a Back End server is employed. The server will retrieve data from our Database and will then provide it in the form of hypertext to the users of the Web Application so that they will have a graphical and user-friendly representation of the given data.

Users interact with the Web Application with the HTTP protocol. In the future, our Web Application will make use of the HTTPS protocol in order to have secure connections between the users and our Web Application.

HTTPS is very important since it encrypts data exchanged by the 2 parts in such a way that it will be impossible for an hacker to sniff (overhear a communication) and retrieve private data contained in the exchanged packets.

In this way it is possible to prevent anyone, apart from the considered user, to get sensitive data by just intercepting the packets.

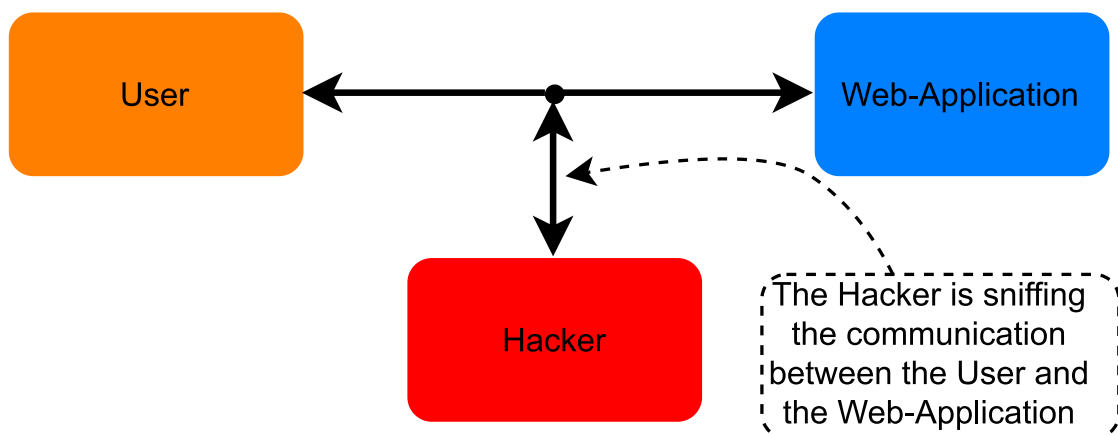


Figure 5.6: An example of an Hacker sniffing the communication between the User and the Web Application.

We provide here below a diagram that represents a very simple example of interaction between a user and our server. In this example the user wants to retrieve all the Shared-Tasks that we keep track of by entering into the *tasks* Web-Page.

As you can see, the request is handled and processed by our server which then retrieves the corresponding data from the database and forwards it back to the user.

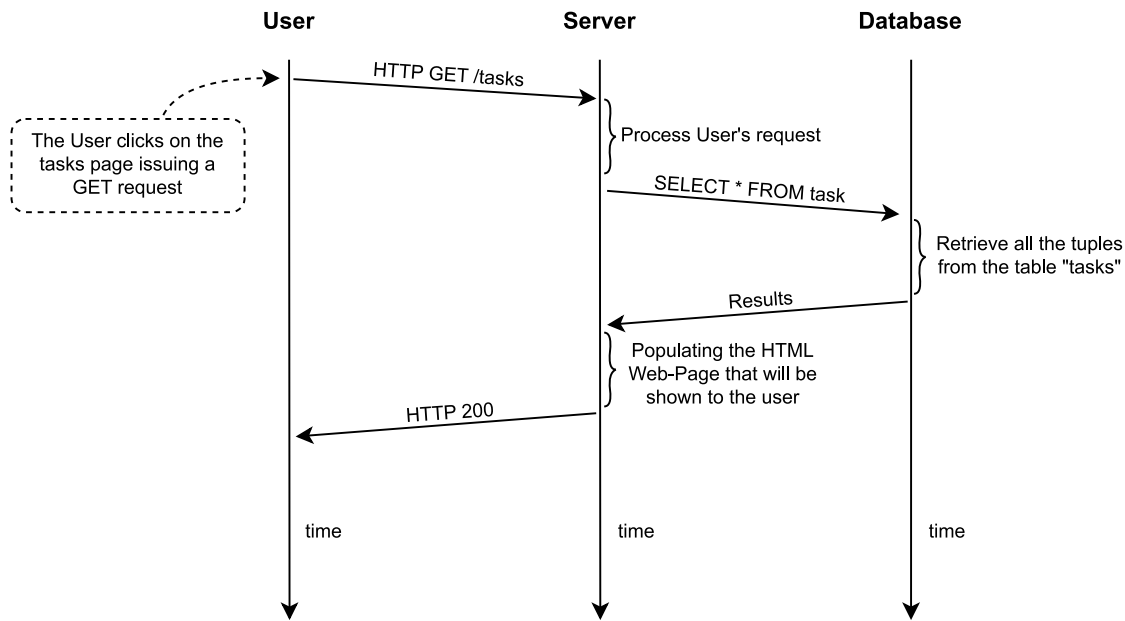


Figure 5.7: A simple example in which it is possible to see the communication between the User, Server and Database.

5.4.3 WEB APPLICATION: SECURITY ISSUES

Since the Web Application is exposed to the Internet, it could be possible that some malicious users try to exploit security vulnerabilities in order to compromise the confidentiality, integrity, or availability of the system or data. When it comes to Web Applications, the most known security vulnerabilities are SQL injections, Cross Site Scripting (XSS) and Cross-Site Requests.

SQL INJECTION

SQL injection is a vulnerability that allows an attacker to execute malicious SQL statements by injecting them into a web application's input fields. Consider the example in which we have a Login form where the user is asked to provide a username and a password. Then the credentials are checked against the ones in the database to check whether they are valid or not. This might be done by the following statement which counts how many accounts have the specified username and password. Then if the number of accounts is at least one, it means that we have a match and we can let the user enter in the private area of our Web Application.

5.4. WEB APPLICATION

```
1 SELECT COUNT(*) FROM account WHERE username="$input.username" and
  password="$input.password";
2
```

Code 5.1: SQL statement to retrieve the number of users having the specified username and password.

As you can see, the input is directly provided to the statement without any sanitization. In fact a malicious user could provide the username " **OR 1=1;#** . In this case the SQL statement expanded with the considered username would be:

```
1 SELECT COUNT(*) FROM account WHERE username="" OR 1=1;# and password=
  "$input.password";
2
```

Code 5.2: SQL statement with SQL injection applied.

As you can see, the condition always evaluates to *True* because the password part in the statement has been commented. In this case the malicious user can Login without having an account!

At this point you may wonder that this is not a big threat, but what if the provided username was " **OR 1=1; DELETE * FROM account;#** ?

```
1 SELECT COUNT(*) FROM account WHERE username="" OR 1=1; DELETE * FROM
  account;# and password="$input.password";
2
```

Code 5.3: SQL statement with SQL injection applied.

In this case all the accounts of our users would be erased!

It is necessary to handle the SQL injection vulnerability with care to avoid these potential issues. This can be done by employing prepared statements.

Prepared statements are very useful against SQL injections because they allow to separate the SQL query logic from the user input, thus preventing malicious code injection. By means of prepared statements the input that is provided by the user is treated as a plain string, therefore even though the user provides malicious code it will not be executed.

CROSS-SITE SCRIPTING

XSS is a vulnerability in which attackers take advantage of the fact that web applications execute scripts on the user's browser.

There are different categories of XSS attacks and they are usually classified as:

- Stored: the code is stored in the database before its execution.
- Reflected: the code is reflected by a server.

To make a simple example, we consider the case in which our Web Application allows users to create their own accounts and view the accounts of other users. Imagine that when visiting a profile you can see the username and the biography of the other account.

In this scenario, a malicious user Bob could provide as his own biography the following one:

```
1 <script>alert("You have been hacked")</script>
2
```

Code 5.4: Example of a Cross-Site Scripting attack.

In that case, when anyone visits Bob's page would see a popup appearing telling him that he has been hacked.

This is due to the fact that when the page is rendered with Bob's biography, the biography would be interpreted as part of the Document Object Model (DOM) and executed by the browser. In this case nothing bad is happening, but the malicious script could be used in more bad ways such as redirecting the user to a malicious Web Page.

To overcome this issue it is possible to perform input sanitization. It consists in performing specific operations at Back End side to inspect the raw input data provided by the user and then transform the input into valid data for our application. There are several techniques that can be employed to sanitize the input such as parsers and regular expressions.

In the example mentioned above, since HTML is not a regular language it is impossible to perform input sanitization with regular expressions. In these cases it is better to use a parser that strips the HTML tags from the input.

CROSS-SITE REQUEST FORGERY

Cross-Site Requests can cause potential vulnerabilities in a Web Application. In particular a malicious user could perform a Cross-Site Request Forgery (CSRF) attack targeting a victim user of our Web Application.

In that case, the victim must be logged into our Web Application. The victim is then attracted by the attacker to a malicious website that exploits the victim's session cookie in order to perform some actions in our Web Application. To put it in simple words, the attacker impersonates the victim exploiting the victim's

5.4. WEB APPLICATION

session cookie.

This problem arises because the browser always attaches the cookies related to a given Website even if a request comes from a different domain. Therefore the server is in principle not able to distinguish whether a request is cross-site or same-site.

A possible solution to overcome this issue is the embedding of a secret token that must be enclosed inside the web pages. This token is set by the server and is specific for every user. In this way, whenever a user submits a request, the server checks the token against the one that it provided before to the user in order to establish if they match or not. If they match then the request can be accepted.

FRONT END INPUT VALIDATION

After having seen that input sanitization at the Back End is very important in order to check the validity of the data provided by the user, we will also discuss now the importance of input validation performed at the Front End.

Validating the input data at the Front End can help in the following ways:

- Provide a better user experience. Errors are caught and displayed to the user immediately.
- Reduce the server load. In fact, it is possible to avoid sending requests to the server if they are not correctly formatted.

We want to emphasize that it is mandatory to validate and sanitize the user input also at the Back End because the Front End input validation can be by-passed easily by a malicious user. Front End input validation must not be considered as a security measure but rather as an optimization for both the server and the users.

6

Implementation of the System

In this chapter we will provide an overview of the implementation of the system.

We will start by showing how the Web Application looks like from a user point of view. The visual design of the Web Application has been made in such a way to be responsive and user friendly.

In addition, we will see some examples of the important configuration files that have been prepared in order to deploy the system.

Finally we will see some screenshots reporting the system running by means of Minikube, which is a tool that can be used in order to test the deployment of a Kubernetes cluster even in a simple personal computer.

6.1 WEB APP IMPLEMENTATION

The Front End part of the Web Application has been implemented by means of HTML, JavaScript and CSS while the Back End part has been implemented by means of Flask and PostgreSQL.

After having designed the database, we decided to make use of PostgreSQL Database Management System (DBMS) because PostgreSQL is a very powerful open-source system that can be used to manage relational databases. It has been developed for many years and it is really stable. It is also employed as the primary data store in many web applications worldwide.

In addition we employed the SQL-Alchemy Python library to interact with our

6.1. WEB APP IMPLEMENTATION

Database. The SQL-Alchemy Python library also allows to create SQL statements by means of prepared statements.

This library can be used as an Object Relational Mapper (ORM) tool that translates our Python classes into corresponding tables on the relational database and automatically converts function calls to SQL statements.

We also ensured to apply valid techniques in order to mitigate the potential security vulnerabilities that were mentioned in the previous chapter in the following ways:

- **SQL Injection:** the SQL-Alchemy Python library allows to create SQL statements by means of prepared statements so that any input provided by the user will be treated as a plain string. In this way, the user cannot execute any operations apart from the ones that are designed for the specific requests.
- **XSS:** HTML tags are stripped with the BeautifulSoup Python library in order to mitigate XSS issues. This methods applies an HTML parser to the provided input since regular expressions do not work for the HTML language.
- **CSRF:** CSRF tokens are applied in order to avoid potential issues due to cross-site requests.

In addition, in our Web Application we validated all the input-fields both at the Back End and at the Front End to enhance both the user experience and the security level.

Lastly, we encrypted the user password through a hashing function. Hashing functions have a really important property that implies that it will not be possible to reverse the hashing function and retrieve the original data.

The Web Application has been designed in order to dynamically resize its content according to the screen-size in such a way that it is visually appealing for both smartphones and computers. In addition, the various components have a responsive design that enhances the user experience while interacting with our Web Application.

Here we provide only some of the images representing how the Web Application looks like from a user perspective.

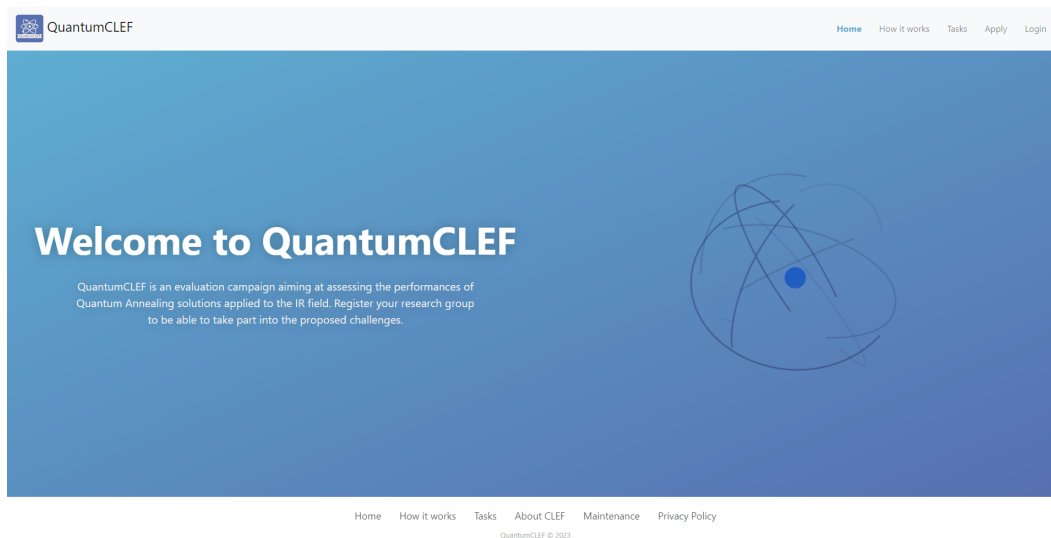


Figure 6.1: The homepage of our Website.

From the homepage it is possible to notice that a guest user can have access to several sections such as the Home section, the Tasks section and so on.

When someone wants to participate in one of the possible tasks, there is a corresponding form that needs to be compiled in the Apply section. In this form a group is required to submit the email that will be used to inform about the acceptance or rejection of the group's participation by the organizers. It is also required to submit the group members, a description of the group that allows organizers to have a better knowledge of the background of the different candidates and the task the group wants to participate in.

Figure 6.2: The form to apply for a task.

6.1. WEB APP IMPLEMENTATION

Once a group has submitted the Application form it has to wait until the organizers will send a corresponding acceptance or rejection email.

In the case of acceptance, the group is asked to confirm the registration providing a username and a password.

The submitted credentials will be used in order to login into the group's protected area, which of course cannot be accessed by other groups since the credentials are not shared with other groups.

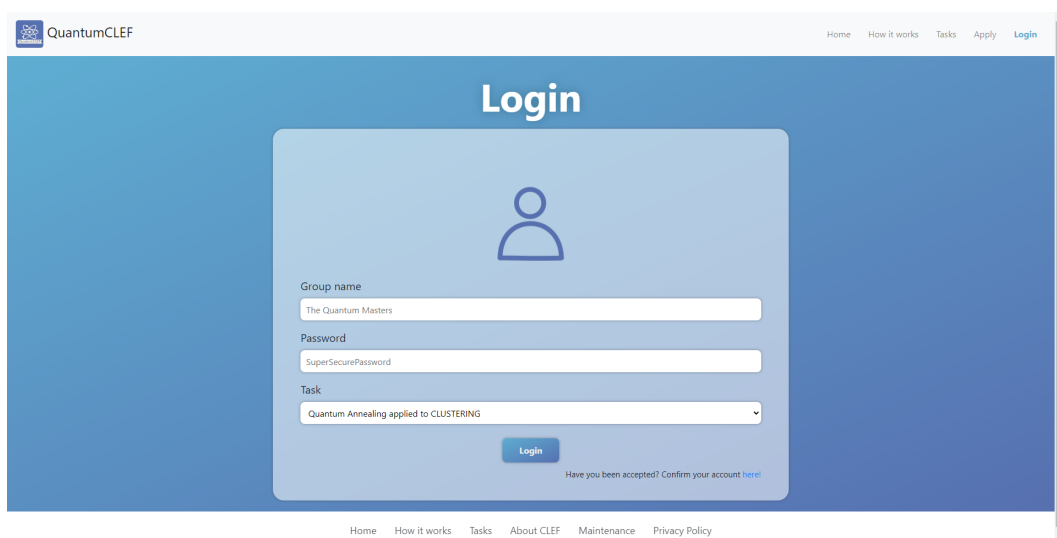
The image shows a web browser window displaying the QuantumCLEF login page. The page has a blue header with the QuantumCLEF logo on the left and navigation links (Home, How it works, Tasks, Apply, Login) on the right. The main content area has a dark blue background with the word "Login" in white. In the center, there is a light blue rounded rectangle containing a user icon and three input fields: "Group name" (with the value "The Quantum Masters"), "Password" (with the value "SuperSecurePassword"), and "Task" (a dropdown menu with "Quantum Annealing applied to CLUSTERING" selected). Below the input fields is a blue "Login" button and a link that says "Have you been accepted? Confirm your account here!". At the bottom of the page, there is a footer with navigation links: Home, How it works, Tasks, About CLEF, Maintenance, and Privacy Policy.

Figure 6.3: The login form to access the protected area.

After having provided the correct credentials, the group can have access to its own protected area. This consists in a Dashboard reporting some useful statistics for the group and its credentials to have access to the machines in our Submission System that will be used for developing the solutions for the given tasks.

As it is possible to see, a group can see how much QPU time it has left according to the amount of QPU time that has been previously chosen by the organizers for each group according to the given tasks.

In addition, the group can see how many submissions it has done so far.

The credentials appear blurred intentionally. It is sufficient to hover the mouse over the blurred areas to view them clearly.

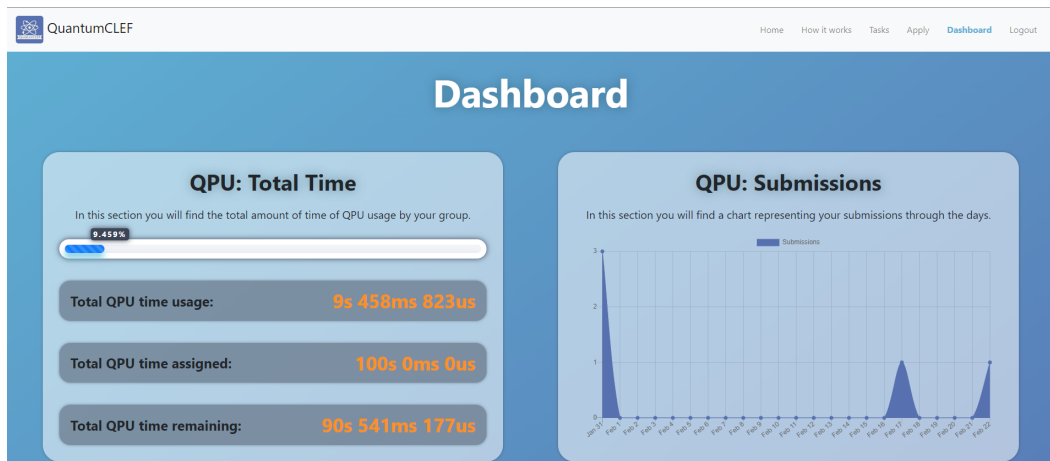


Figure 6.4: A first view into the groups' protected area.

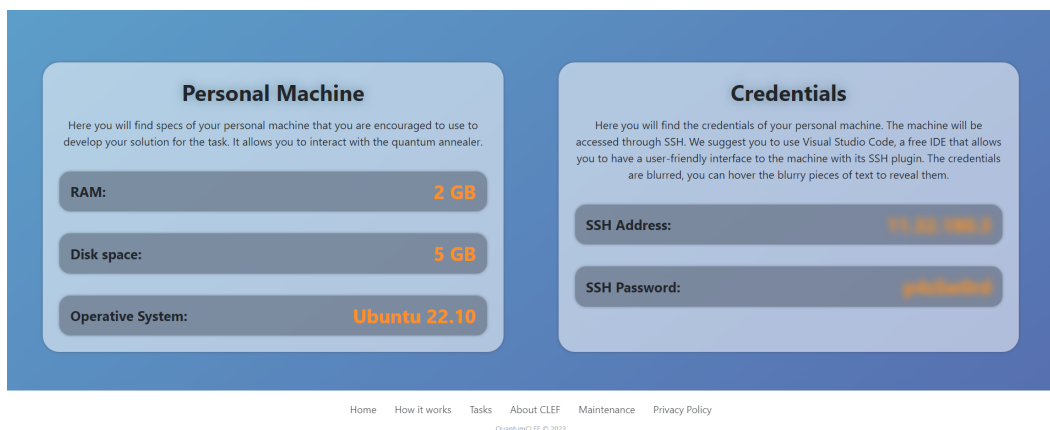


Figure 6.5: A second view into the groups' protected area.

An administrator/organizer has a personal protected area. In that area it is possible to have some insights regarding how the tasks are currently going and it is also possible to add or modify the tasks.

Furthermore, we allow the administrator to view the each group's data and the corresponding submissions. This is done in order to let the administrator accept or reject group applications or handle special requests from groups (e.g., possibility of adding additional QPU time for a group if needed).

Here we provide a brief insight about how the administrator views the different groups that are currently subscribed and participating to the active tasks in a given moment.

6.2. THE SUBMISSION SYSTEM FROM A KUBERNETES POINT OF VIEW

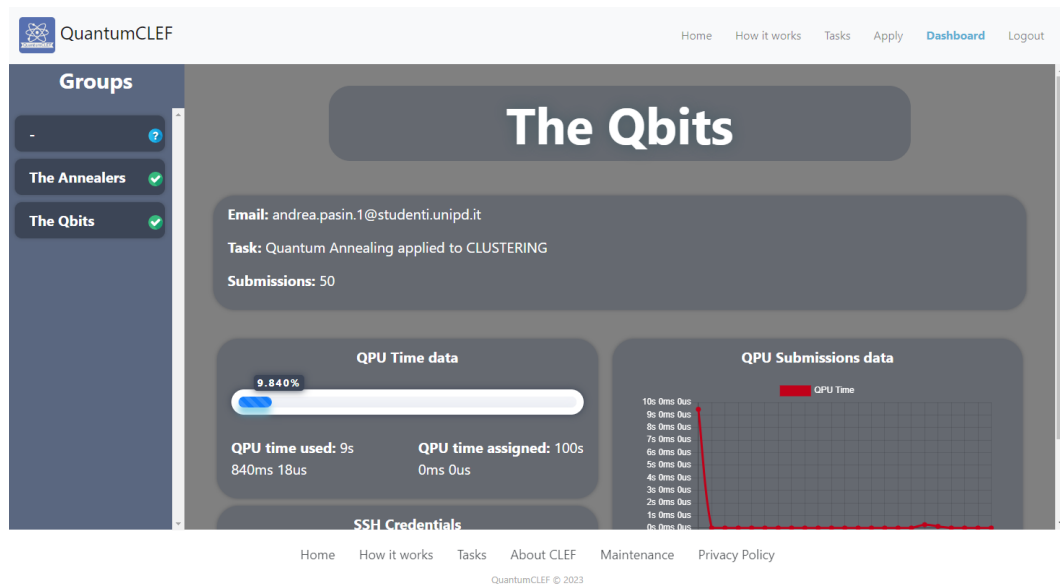


Figure 6.6: A view into the administrator section where it is possible to have a look at the groups who have applied to the currently active tasks and modify the corresponding data.

6.2 THE SUBMISSION SYSTEM FROM A KUBERNETES POINT OF VIEW

Here we will report and describe how each component of our system has been deployed according to the Kubernetes objects. First of all, we provide here a schema representing the main objects employed to build our Submission System according to Kubernetes.

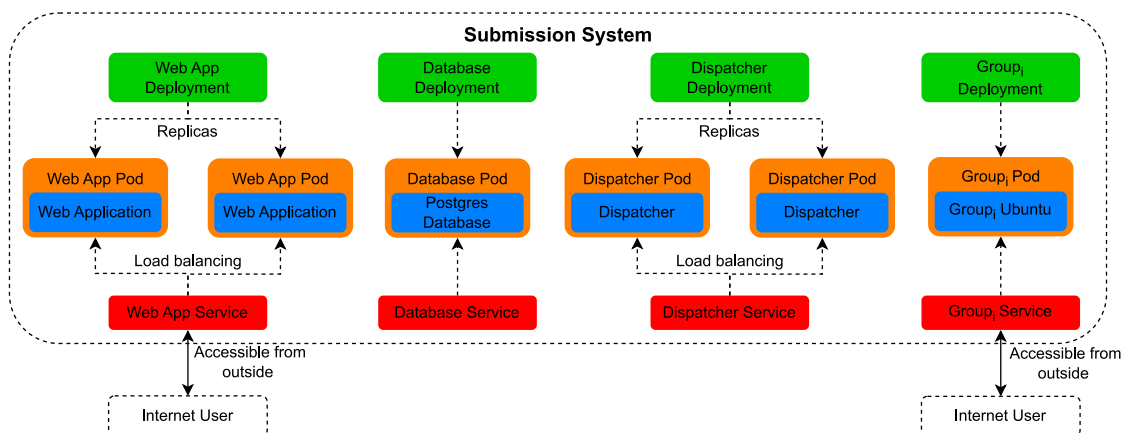


Figure 6.7: The representation of our Submission System based on some of the most important Kubernetes objects.

It is possible to see that each container must be deployed inside a Pod. Then, based on each application that we need to deploy, we have a Deployment that creates the opportune pods and a Service that is used in order to refer to the deployed application in a high-level way.

We want to highlight here the importance of services. In fact, each Pod can be deployed in principle in any node that is part of our cluster. We recall that with the word *node* we refer to a physical machine.

This means that Kubernetes decides where to deploy a given Pod and which IP address it has. In addition, if a Pod fails, Kubernetes will automatically restart it possibly deploying it to a new node and assigning it a new IP address.

As a consequence, in this scenario we would need to constantly keep track of all the pods' IP addresses in order to communicate with them. A Service component is fundamental to avoid this difficult job because it provides a way to communicate with a given application by automatically keeping track of where the pods are located!

From the presented schema we can derive the following characteristics:

- The Web Application Back End needs to be accessed from outside the cluster. This can be easily done by means of a Service simply assigning some specific properties in the corresponding Service *yaml* file. In this case, since the Web Application is likely to be accessed by many users, we decided to deploy 2 replicas of the Web Application. According to a load balancing mechanisms provided by services we are able to redirect users to one of the two considered replicas according to the workload that each replica has at a given time.
- The Database and Dispatcher are not accessible from outside to make them more secure.
- Each group container will be accessible from outside and we have only 1 replica per container. This is due to the fact that even if a failure happens, the container will be restarted automatically by Kubernetes and each group will access to its own container which implies that it is unlikely to have a high workload in a single group container.

In addition to the components present in the schema, we want to highlight that we employed also objects of types PersistentVolume, PersistentVolumeClaim and Secret.

Having a persistent storage is fundamental for stateful applications in Kubernetes. In fact, we want that whenever the container is stopped or fails, it will be restarted by Kubernetes with its previous state. In our case, it is fundamental

6.2. THE SUBMISSION SYSTEM FROM A KUBERNETES POINT OF VIEW

to have persistent storage for all the users' containers and for the database. In addition, Secret objects have been used to store the Database credentials and the Web Application Backend secret keys that are used for security purposes. Here we report as an example the yaml code that is used to deploy the database in the Kubernetes cluster.

```
1 apiVersion: apps/v1
2 # The kind of this object is Deployment
3 kind: Deployment
4 metadata:
5   name: postgres-deployment
6   labels:
7     app: postgresdb
8 spec:
9   replicas: 1
10  selector:
11    matchLabels:
12      app: postgresdb
13  template:
14    # The PostgreSQL Pod
15    metadata:
16      labels:
17        # The Pod label
18        app: postgresdb
19    spec:
20      containers:
21        - name: postgresdb
22          # Specifying the Docker image
23          image: postgres
24          ports:
25            # The postgres' associated port
26            - containerPort: 5432
27          env:
28            # The root password
29            - name: POSTGRES_PASSWORD
30              valueFrom:
31                secretKeyRef:
```

```

32         # The kubernetes secret containing our data
33         name: postgres-secret
34         # The key from which the corresponding value
will be retrieved
35         key: postgres-root-password
36     # The root username
37     - name: POSTGRES_USER
38       valueFrom:
39         secretKeyRef:
40           name: postgres-secret
41           key: postgres-root-username
42     # The database name
43     - name: POSTGRES_DB
44       valueFrom:
45         secretKeyRef:
46           name: postgres-secret
47           key: postgres-database-name
48     # Referring to a PersistentVolume object we
created with another file
49     volumeMounts:
50     - name: postgres-persistent-storage
51       mountPath: /var/lib/postgres
52     # Referring to a PersistentVolumeClaim object we
created with another file
53     volumes:
54     - name: postgres-persistent-storage
55       persistentVolumeClaim:
56         claimName: postgres-pv-claim
57 ---
58 apiVersion: v1
59 kind: Service
60 metadata:
61   name: postgresdb-service
62 spec:
63   selector:
64     # This matches the Pod label

```

6.3. AN IN-DEPTH VIEW INTO EACH CONTAINER

```
65   app: postgresdb
66   ports:
67     - protocol: TCP
68       port: 5432
69       targetPort: 5432
70
```

Code 6.1: The Kubernetes yaml code for the Deployment and Service of the PostgreSQL database.

6.3 AN IN-DEPTH VIEW INTO EACH CONTAINER

As we already explained, each application is run inside a container. Therefore, before deploying the Kubernetes system it is required to create the opportune container images from which Kubernetes will create the builds inside the pods.

A container image is a lightweight and executable package which includes all the necessary components required to run an application. It is similar to a Virtual Machine image, but it is optimized for running only the given application as a container. Container images provide a portable way to package and distribute applications. In fact, they can be easily moved across different environments without requiring any additional configuration.

In our case we decided to create our own images with Docker. Creating an image with Docker is quite easy and it can be done by means of a *Dockerfile*, which is a file where it is possible to specify how the image should be created with a specific syntax.

THE WEB APPLICATION

The Web Application container image that we created uses as basis the Python image[58]. Using the Python base image, we then install our specific packages that are used in order to make the Web Application Back End up and running (e.g. Flask). In addition, since the Web Application needs to communicate with other parties that are located outside the container, we expose a specific port that will be used to exchange messages outside the container itself. We issue then a specific command that allows to start the Web Application once the container is started.

THE DATABASE

The Database container image that we created uses as basis the PostgreSQL image [59]. This requires us to provide some environment variables for the image itself which are the username and password of the user that owns the server. In addition we need to specify the name of the database that will store our tables and data in general. These environment variables will be then passed by means of an opportune Kubernetes Secret object once the container is deployed.

THE DISPATCHER

The Dispatcher container image that we created uses as basis the Python image [58]. Using the Python base image we then install our specific packages that are used in order to make the Dispatcher work. As for the Web Application container, we expose a specific port that will be used to exchange messages outside the container itself. Also in this case, a specific command that allows to start the Dispatcher once the container starts is issued.

THE GROUP'S WORKSPACE

The Group's Workspace container image that we created uses as basis the Ubuntu 22.10 image [60]. The image is then customized by changing the credentials of the root user and adding a corresponding user that does not have administrator privileges. This user will be the one that the considered group will use to develop their own application through an installed OpenSSH server. Also in this case, we expose a port to allow the communication with the container from outside. In addition, we also install the required Python libraries and our own custom Python library that allows users to interact with the Dispatcher issuing requests of solving a specific problem.

Once all the images are created, Kubernetes will use them to deploy the corresponding applications with its own objects. As you can guess, creating all the Group container images is done only once all the groups applied for the tasks while the other images (the Web Application, Database and Dispatcher) are created only once for the whole lifetime of the QuantumCLEF campaign. In particular, the Group container image is created with a Dockerfile that uses some variables so that the entire process of creating one image for each group can be done automatically without having an administrator/organizer writing

6.3. AN IN-DEPTH VIEW INTO EACH CONTAINER

the files by itself.

Here we provide as an example the Dockerfile that can be used in order to create the container image for a given group.

```
1 FROM ubuntu:22.10
2 # The variables that can be passed when building the image
3 ARG groupname
4 ARG grouppsw
5 ARG rootpsw
6
7 RUN echo 'APT::Install-Suggests "0";' >> /etc/apt/apt.conf.d/00-
  docker
8 RUN echo 'APT::Install-Recommends "0";' >> /etc/apt/apt.conf.d/00-
  docker
9 # Install important packages
10 RUN DEBIAN_FRONTEND=noninteractive \
11 apt-get update \
12 && apt-get install -y python3 \
13 && apt-get install -y python3-pip \
14 && apt-get -y install vim \
15 && apt install sudo \
16 && apt-get install -y openssh-server \
17 && apt-get install -y systemd \
18 && rm -rf /var/lib/apt/lists/*
19
20 # Change the root password
21 RUN echo "root:$rootpsw" | chpasswd
22
23 # Add a user called <groupname> with password <grouppsw> as NON
  superuser
24 RUN useradd -m $groupname && echo "$groupname:$grouppsw" | chpasswd
25
26 # Create a quantumannealing work directory to install our package
  that is used to communicate with the Dispatcher
27 WORKDIR /quantumannealing
28 COPY requirements.txt /quantumannealing/requirements.txt
29 RUN pip install -r requirements.txt --src /usr/local/src
30
31 # Copy our own package to access the quantum annealer
32 COPY . .
33 WORKDIR /quantumannealing/package
34 RUN pip install .
35
36 # Expose port 22 to allow SSH access
37 EXPOSE 22
38
39 # Start SSH server
40 RUN service ssh start
```



```
41 CMD ["/usr/sbin/sshd", "-D"]
```

```
42
```

Code 6.2: The Dockerfile used to create the container image for the groups.

To build the image it is sufficient to run the following command (or use our Python script that automates the process):

```
1 # builds the image with name <group name>, creates a username and
  # password for the group and a password for the root
2 docker build . -t <group name> --build-arg groupname=<group name> --
  build-arg grouppsw=<group password> --build-arg rootpsw=<root
  password>
```

Code 6.3: The command to build the Docker image for the group's workspace container.

6.4 THE SUBMISSION SYSTEM RUNNING

To test if the Submission System is working correctly in a easy and practical way, we employed Minikube.

Minikube is an open-source tool that allows to run a Kubernetes cluster locally on a personal computer. It has been created in order to facilitate the developing and testing of Kubernetes applications without requiring to have access to a full-scale cluster.

It provides a lightweight Kubernetes runtime environment, including all the necessary components. One of the main advantages of using Minikube is that it provides an isolated environment for developing and testing Kubernetes applications without affecting other environments on the host machine.

In the image provided below, it is possible to see our application in terms of deployments.

Here it is also possible to see that we added for test purposes another deployment which is pgAdmin that allows us to check and manage the state of our database in a visual way.

6.4. THE SUBMISSION SYSTEM RUNNING

```
PS C:\Users\andre> kubectl get deployment
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
dispatcher-deployment	1/1	1	1	39s
flask-deployment	1/1	1	1	4m33s
group1-deployment	1/1	1	1	99s
pg-admin	1/1	1	1	6m47s
postgres-deployment	1/1	1	1	6m51s

Figure 6.8: The Kubernetes deployments of our Submission System.

Here below we can see what a participant should see by making use of the Visual Studio Code IDE. In this screenshot it is possible to see an example of a problem defined by means of two QUBOs that will be submitted with our own library that is installed in each group's workspace.

From the terminal that is located in the lower part of the image, we can see how the results look like.

```
1 from clef.qa import qa_access
2 Q1 = {(0,0): 1, (1,1): 1, (2,3): 2, (1,2): -2, (0,3): -2}
3 Q2 = {(0,0): 1, (1,1): 1, (2,3): 2, (1,2): -2, (0,3): -2}
4 solutions=qa_access.submit_qubos([Q1,Q2],[2,2],num_reads=2)
5 print(solutions)
```

```
$ python3 test.py
None
$ python3 test.py
[SampleSet(rec.array([[0, 1, 1, 0], [-1, 2, 0, 0]]),
dtype=[('sample', 'i1', (4,)), ('energy', '<f8')], ('num_occurrences', '<i8'), ('chain_break_fraction', '<f8')), Variables([0, 1, 2, 3]), {'timing': ('qpu_sampling_time': 139.24, 'qpu_anneal_time_per_sample': 20.0, 'qpu_readout_time_per_sample': 54.55, 'qpu_access_time': 15989.81, 'qpu_access_overhead_time': 1321.19, 'qpu_programming_time': 15759.57, 'qpu_delay_time_per_sample': 20.54, 'total_post_processing_time': 1928.0, 'post_processing_overhead_time': 1928.0), 'problem_id': 'b71fe62-d7e7-4118-8a67-d388d6a47b1b', 'embedding_context': {'embedding': {'2': [3232], '1': [1471], '3': [1531], '0': [1546]}, 'chain_break_method': 'majority_vote', 'embedding_parameters': {}, 'chain_strength': 2}], 'BINARY'), SampleSet(rec.array([[1, 0, 0, 1], [-1, 1, 0, 0]], dtype=[('sample', 'i1', (4,)), ('energy', '<f8')], ('num_occurrences', '<i8'), ('chain_break_fraction', '<f8')), Variables([0, 1, 2, 3]), {'timing': ('qpu_sampling_time': 165.8, 'qpu_anneal_time_per_sample': 20.0, 'qpu_readout_time_per_sample': 42.36, 'qpu_access_time': 15925.37, 'qpu_access_overhead_time': 3896.63, 'qpu_programming_time': 15759.57, 'qpu_delay_time_per_sample': 20.54, 'total_post_processing_time': 373.0, 'post_processing_overhead_time': 373.0), 'problem_id': '4d836c7e-b9ec-4b27-a10e-54d273a922e', 'embedding_context': {'embedding': {'2': [1869], '1': [4750], '3': [1868], '0': [4644]}, 'chain_break_method': 'majority_vote', 'embedding_parameters': {}, 'chain_strength': 2}], 'BINARY')]
```

Figure 6.9: The interface that each group has representing its corresponding workspace if using the Visual Studio Code IDE.

Finally here we see how the group's private area in the Web Application has been modified according to the submission. The dispatcher in fact keeps track of all the submissions sent by the participants.

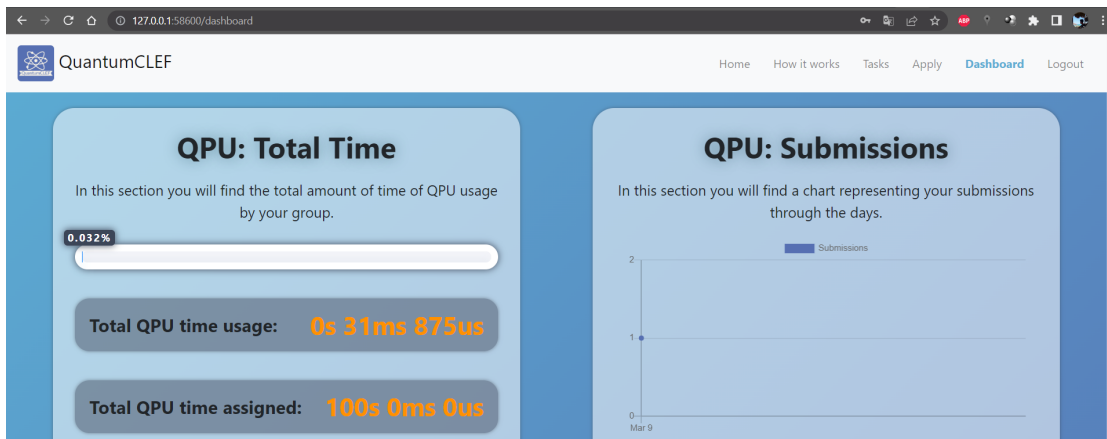


Figure 6.10: The group’s private area where all the submissions are tracked by the Dispatcher.



Conclusions and Future Works

In this work we have seen what we mean by Information Retrieval and why this field is so important in our everyday life. We have had an overview of the importance of both efficiency and effectiveness in IR systems and we understood that effectiveness is much more complex than efficiency since it is a subjective concept. To overcome this issue, evaluation campaigns that follow the Cranfield paradigm are conducted.

Furthermore, we have explored the field of Quantum Computing with a specific insight regarding the Quantum Annealing paradigm that allows us to solve problems that can be formulated as Quantum Unconstrained Binary Optimization problems. We have investigated how a quantum annealer works taking into considerations the quantum annealer devices provided by D-Wave.

Moreover, we have proposed some feasible Shared Tasks that can be carried out in a future QuantumCLEF evaluation campaign. This campaign could be very important to assess whether Quantum Annealing can make a difference in the Information Retrieval field in terms of both effectiveness and efficiency.

Finally, we have designed a Submission System that can be employed to handle the participants of the given tasks. This system is a Distributed System that has been created in order to satisfy the principles of availability, fault-tolerance and security.

In this work we had the opportunity to use and test cutting-edge technologies in the field of Quantum Computing. We understood that Quantum Computing

technologies are not just theoretical and impractical concepts but instead they are becoming more and more powerful and it is very likely that they will have an enormous impact on many fields.

With this work we want to create the opportunity to explore and raise the awareness of the importance of Quantum Annealing for the research community. In fact, with the QuantumCLEF evaluation campaign it will be possible to involve many researchers coming from different research fields who will put their effort on improving the methods and techniques to solve problems with quantum computing devices.

This will benefit both researchers that will be able to understand and use new technologies but also the Information Retrieval field, because only through evaluation campaigns it is possible to assess whether some solutions are more effective than others.

We will continue to improve what we have done so far and we will finally propose the QuantumCLEF evaluation campaign as an evaluation campaign carried out in accordance with CLEF so that a vast number of research groups will be invited to participate.

The evaluation campaign will make use of the resources provided by CINECA [61], which is one of the most important computing centers worldwide. In fact, we already managed to obtain their resources through an agreement. We have been given High Performance Computing resources for emulation purposes (Leonardo [62]) and Quantum Annealing resources.

This campaign will be started in 2024 and we have high hopes that this could help to bring out the true potential of Quantum Computing technologies allowing researchers to work together forming a big research community around this innovative field that still needs to be explored a lot.

References

- [1] Mark Sanderson and W Bruce Croft. "The history of information retrieval research". In: *Proceedings of the IEEE 100.Special Centennial Issue* (2012), pp. 1444–1451.
- [2] Michael K Buckland. "Emanuel Goldberg, electronic document retrieval, and Vannevar Bush's Memex". In: *Journal of the American Society for Information Science* 43.4 (1992), pp. 284–294.
- [3] *Emanuel Goldberg and his Statistical Machine, 1927*. <https://people.ischool.berkeley.edu/~buckland/statistical.html>. Accessed: 2023-01-07.
- [4] Vannevar Bush. "As we may think". In: *interactions* 3.2 (1996), pp. 35–46.
- [5] Mortimer Taube, Cloyd D Gull, and Irma S Wachtel. "Unit terms in coordinate indexing". In: *Journal of the American Society for Information Science* 3.4 (1952), p. 213.
- [6] Hans Peter Luhn. "A statistical approach to mechanized encoding and searching of literary information". In: *IBM Journal of research and development* 1.4 (1957), pp. 309–317.
- [7] Joseph John Rocchio Jr. "Relevance feedback in information retrieval". In: *The SMART retrieval system: experiments in automatic document processing* (1971).
- [8] Martin F Porter. "An algorithm for suffix stripping". In: *Program* 14.3 (1980), pp. 130–137.
- [9] Julie Beth Lovins. "Development of a stemming algorithm." In: *Mech. Transl. Comput. Linguistics* 11.1-2 (1968), pp. 22–31.

REFERENCES

- [10] Gerard Salton, Anita Wong, and Chung-Shu Yang. "A vector space model for automatic indexing". In: *Communications of the ACM* 18.11 (1975), pp. 613–620.
- [11] Stephen Robertson, Hugo Zaragoza, et al. "The probabilistic relevance framework: BM25 and beyond". In: *Foundations and Trends® in Information Retrieval* 3.4 (2009), pp. 333–389.
- [12] Norbert Fuhr. "Optimum polynomial retrieval functions based on the probability ranking principle". In: *ACM Transactions on Information Systems (TOIS)* 7.3 (1989), pp. 183–204.
- [13] Yongfeng Zhang et al. "Towards conversational search and recommendation: System ask, user respond". In: *Proceedings of the 27th acm international conference on information and knowledge management*. 2018, pp. 177–186.
- [14] Sagar Uprety, Dimitris Gkoumas, and Dawei Song. "A survey of quantum theory inspired approaches to information retrieval". In: *ACM Computing Surveys (CSUR)* 53.5 (2020), pp. 1–39.
- [15] Ellen M Voorhees. "The philosophy of information retrieval evaluation". In: *Evaluation of Cross-Language Information Retrieval Systems: Second Workshop of the Cross-Language Evaluation Forum, CLEF 2001 Darmstadt, Germany, September 3–4, 2001 Revised Papers 2*. Springer. 2002, pp. 355–370.
- [16] Monika Arora, Uma Kanjilal, and Dinesh Varshney. "Evaluation of information retrieval: precision and recall". In: *International Journal of Indian Culture and Business Management* 12.2 (2016), pp. 224–236.
- [17] George Hripcsak and Adam S Rothschild. "Agreement, the f-measure, and reliability in information retrieval". In: *Journal of the American medical informatics association* 12.3 (2005), pp. 296–298.
- [18] Kazuaki Kishida. *Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments*. National Institute of Informatics Tokyo, Japan, 2005.
- [19] W Bruce Croft, Donald Metzler, and Trevor Strohman. *Search engines: Information retrieval in practice*. Vol. 520. Addison-Wesley Reading, 2010.
- [20] *The size of the World Wide Web (The Internet)*. <https://www.worldwidewebsite.com/>. Accessed: 2023-02-14.

- [21] Han Zhang et al. "Towards personalized and semantic retrieval: An end-to-end solution for e-commerce search via embedding learning". In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2020, pp. 2407–2416.
- [22] Michael S Lew et al. "Content-based multimedia information retrieval: State of the art and challenges". In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 2.1 (2006), pp. 1–19.
- [23] Phillip Kaye, Raymond Laflamme, and Michele Mosca. *An introduction to quantum computing*. OUP Oxford, 2006.
- [24] John E Hopcroft, Rajeev Motwani, and Jeffrey D Ullman. "Introduction to automata theory, languages, and computation". In: *Acm Sigact News* 32.1 (2001), pp. 60–65.
- [25] James Carlson et al. *The millennium prize problems*. American Mathematical Soc., 2006.
- [26] Peter W Shor. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer". In: *SIAM review* 41.2 (1999), pp. 303–332.
- [27] Eleanor G Rieffel and Wolfgang H Polak. *Quantum computing: A gentle introduction*. MIT Press, 2011.
- [28] *Superposition and Entanglement*. <https://www.quantum-inspire.com/kbase/superposition-and-entanglement/>. Accessed: 2023-02-01.
- [29] Olivier Carnal and Jürgen Mlynek. "Young's double-slit experiment with atoms: A simple atom interferometer". In: *Physical review letters* 66.21 (1991), p. 2689.
- [30] *GATE MODEL QUANTUM COMPUTING VERSUS QUANTUM ANNEALING - THE PRACTICAL AND MATHEMATICAL DIFFERENCES*. <https://www.mvpbakery.io/blog/gate-model-quantum-computing-versus-quantum-annealing-the-practical-and-mathematical-differences>. Accessed: 2023-01-16.
- [31] *What is Quantum Annealing?* https://docs.dwavesys.com/docs/latest/c_gs_2.html. Accessed: 2023-01-14.
- [32] Antonia Creswell et al. "Generative adversarial networks: An overview". In: *IEEE signal processing magazine* 35.1 (2018), pp. 53–65.

REFERENCES

- [33] Dor Bank, Noam Koenigstein, and Raja Giryes. “Autoencoders”. In: *arXiv preprint arXiv:2003.05991* (2020).
- [34] Christian Ledig et al. “Photo-realistic single image super-resolution using a generative adversarial network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4681–4690.
- [35] *White paper: Computational Power Consumption and Speedup*. <https://www.dwavesys.com/resources/white-paper/computational-power-consumption-and-speedup/>. Accessed: 2023-01-14.
- [36] Francesco Bova, Avi Goldfarb, and Roger G Melko. “Commercial applications of quantum computing”. In: *EPJ quantum technology* 8.1 (2021), p. 2.
- [37] Kyung-Kyu Ko and Eun-Sung Jung. “Development of cybersecurity technology and algorithm based on quantum computing”. In: *Applied Sciences* 11.19 (2021), p. 9085.
- [38] Marcos Lopez de Prado. “Generalized optimal trading trajectories: a financial quantum computing application”. In: *Available at SSRN 2575184* (2015).
- [39] Nicola Ferro and Carol Peters. “Information Retrieval Evaluation in a Changing World Lessons Learned from 20 Years of CLEF”. In: (2019).
- [40] Alberto Barrón-Cedeno et al. “Report on the 13th Conference and Labs of the Evaluation Forum (CLEF 2022) Experimental IR Meets Multilinguality, Multimodality, and Interaction”. In: *ACM SIGIR Forum*. Vol. 56. 2. ACM New York, NY, USA. 2023, pp. 1–15.
- [41] Maurizio Ferrari Dacrema et al. “Towards feature selection for ranking and classification exploiting quantum annealers”. In: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2022, pp. 2814–2824.
- [42] Fred Glover, Gary Kochenberger, and Yu Du. “A tutorial on formulating and using QUBO models”. In: *arXiv preprint arXiv:1811.11538* (2018).
- [43] IBM ILOG Cplex. “V12. 1: User’s Manual for CPLEX”. In: *International Business Machines Corporation* 46.53 (2009), p. 157.

- [44] Riccardo Nembrini, Maurizio Ferrari Dacrema, and Paolo Cremonesi. “Feature selection for recommender systems with quantum computing”. In: *Entropy* 23.8 (2021), p. 970.
- [45] *D-Wave NetworkX*. <https://docs.ocean.dwavesys.com/projects/dwave-networkx/en/latest/>. Accessed: 2023-01-14.
- [46] T Soni Madhulatha. “An overview on clustering methods”. In: *arXiv preprint arXiv:1205.1117* (2012).
- [47] Kenichi Kurihara, Shu Tanaka, and Seiji Miyashita. “Quantum annealing for clustering”. In: *arXiv preprint arXiv:1408.2035* (2014).
- [48] Hartmut Neven et al. “Training a binary classifier with the quantum adiabatic algorithm”. In: *arXiv preprint arXiv:0811.0416* (2008).
- [49] Edward Boyda et al. “Deploying a quantum annealing processor to detect tree cover in aerial imagery of California”. In: *PloS one* 12.2 (2017), e0172505.
- [50] Hartmut Neven et al. “Qboost: Large scale classifier training with adiabatic quantum optimization”. In: *Asian Conference on Machine Learning*. PMLR, 2012, pp. 333–348.
- [51] Christopher JC Burges. “From ranknet to lambdarank to lambdamart: An overview”. In: *Learning* 11.23-581 (2010), p. 81.
- [52] Tao Qin and Tie-Yan Liu. “Introducing LETOR 4.0 datasets”. In: *arXiv preprint arXiv:1306.2597* (2013).
- [53] Sören Becker et al. “Interpreting and Explaining Deep Neural Networks for Classification of Audio Signals”. In: *CoRR* abs/1807.03418 (2018). arXiv: 1807.03418.
- [54] Derek Greene and Pádraig Cunningham. “Practical solutions to the problem of diagonal dominance in kernel document clustering”. In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 377–384.
- [55] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: <http://archive.ics.uci.edu/ml>.
- [56] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

REFERENCES

- [57] Kyoung-Taek Seo et al. "Performance comparison analysis of linux container and virtual machine for building cloud". In: *Advanced Science and Technology Letters* 66.105-111 (2014), p. 2.
- [58] *Python Docker image*. https://hub.docker.com/_/python. Accessed: 2023-02-01.
- [59] *PostgreSQL DOcker image*. https://hub.docker.com/_/postgres. Accessed: 2023-02-01.
- [60] *Ubuntu Docker image*. https://hub.docker.com/_/ubuntu. Accessed: 2023-02-01.
- [61] *CINECA*. <https://www.cineca.it/>. Accessed: 2023-02-01.
- [62] *LEONARDO SUPERCOMPUTER PRE-EXASCALE*. <https://leonardo-supercomputer.cineca.eu/it/home-it/>. Accessed: 2023-02-01.

Acknowledgments

I would like to express my deepest gratitude to professor Nicola Ferro, who gave me the opportunity of participating in this promising project that might have a very powerful impact for the Information Retrieval field in the future. He has been a source of inspiration to me.

In addition, I would like to thank professors Maurizio Ferrari Dacrema and Paolo Cremonesi from Politecnico of Milan for their precious support and for the provided material that has been helpful to better understand how Quantum Annealing works.