Quantum Annealing for Machine Learning: Applications in Feature Selection, Instance Selection, and Clustering

Notebook for the QuantumCLEF Lab at CLEF 2025

Chloe Pomeroy[1,†], Aleksandar Pramov[1,†], Karishma Thakrar[1,†] and Lakshmi Yendapalli[1,*,†]

[1]Georgia Institute of Technology, North Ave NW, Atlanta, GA 30332

Abstract

This paper explores the applications of quantum annealing (QA) and classical simulated annealing (SA) to a suite of combinatorial optimization problems in machine learning, namely feature selection, instance selection, and clustering. We formulate each task as a Quadratic Unconstrained Binary Optimization (QUBO) problem and implement both quantum and classical solvers to compare their effectiveness. For feature selection, we propose several QUBO configurations that balance feature importance and redundancy, showing that quantum annealing (QA) produces solutions that are computationally more efficient. In instance selection, we propose a few novel heuristics for instance-level importance measures that extend existing methods. For clustering, we embed a classical-to-quantum pipeline, using classical clustering followed by QUBO-based medoid refinement, and demonstrate consistent improvements in cluster compactness and retrieval metrics. Our results suggest that QA can be a competitive and efficient tool for discrete machine learning optimization, even within the constraints of current quantum hardware.

Keywords

Quantum annealing, Simulated annealing, QUBO formulation, Feature selection, Instance selection, Clustering, DWave Quantum Annealer

1. Introduction

As machine learning systems are applied to ever-larger datasets, the demands placed on core workflows like feature selection, instance selection, and clustering have grown accordingly. These tasks often involve complex, combinatorial decisions that are challenging to solve efficiently, especially as feature spaces expand into the thousands and datasets span millions of instances. In many cases, classical algorithms struggle to keep up, either becoming computationally prohibitive or falling back on heuristics that don't guarantee globally optimal solutions.

In response to these challenges, there has been growing interest in leveraging quantum computing paradigms, particularly quantum annealing (QA), for machine learning optimization tasks. By formulating these as Quadratic Unconstrained Binary Optimization (QUBO) problems or Ising models, QA can be applied to select optimal subsets of features or instances, or to identify meaningful clusters. QA offers a fundamentally different mechanism for exploring solution spaces by exploiting quantum tunneling, potentially enabling it to escape local minima more effectively than classical counterparts like simulated annealing (SA). With commercial quantum annealers, such as those provided by D-Wave Systems, now accessible to researchers, it is possible to empirically explore the strengths and limitations of QA in practical machine learning contexts.

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain

*Corresponding author.

†These authors contributed equally.

cpomeroy6@gatech.edu (C. Pomeroy); apramov3@gatech.edu (A. Pramov); kthakrar3@gatech.edu (K. Thakrar); ryendapalli3@gatech.edu (L. Yendapalli)

0009-0004-2283-7909 (C. Pomeroy); 0009-0005-9049-1337 (A. Pramov); 0009-0008-2563-7370 (K. Thakrar); 0009-0001-8486-4804 (L. Yendapalli)

The 2025 edition of the Quantum CLEF Competition investigates the feasibility of performing traditional machine learning (ML) tasks by using quantum annealers and comparing their performance to classical methods. It features three subtasks, each to be solved with algorithms runs using both Quantum Annealing (QA) and a Simulated Annealing (SA). Task 1 (Feature Selection) involves selecting the smallest set of features that preserves performance for learning-to-rank on benchmark web collections (MQ2007, ISTELLA) and for an item-based k-NN recommender on a private music corpus with 100- and 400-dimensional item-content matrices. Task 2 (Instance Selection) targets cost-effective fine-tuning of an LLM (Llama 3.1) for sentiment classification by reducing training instances from the Vader NYT and Yelp Reviews datasets without degrading F1 score. Finally, Task 3 (Clustering) requires generating centroid embeddings for the ANTIQUE question-answer corpus, evaluated using the Davies–Bouldin index and query-time nDCG@10 to assess how clustering can accelerate downstream information retrieval tasks [1, 2].

In this work, we investigate the use of quantum annealing for the three aforementioned core machine learning tasks. We formulate each task as a QUBO problem, suitable for execution on D-Wave's Advantage_System quantum annealer. To assess the comparative performance of QA, we also implement classical simulated annealing (SA) using D-Wave's classical solvers. While quantum annealing remains in its early stages and current hardware imposes certain constraints (e.g., limited qubit connectivity, noise, problem size), our findings show that QA produces competitive solutions and serves as a promising component in hybrid ML pipelines. This work contributes to the growing body of research on the practical viability of quantum optimization for real-world machine learning challenges. All code used in this study is available at the respective GitHub repository for each of the three application areas explored in this work: Feature Selection (https://github.com/dsgt-arc/qclef-2025-feature), Instance Selection (https://github.com/dsgt-arc/qclef-2025-instance), and Clustering (https://github.com/dsgt-arc/qclef-2025-clustering).

2. Related Work

Quantum annealing (QA) has emerged as a promising approach for solving combinatorial optimization problems by leveraging quantum fluctuations to escape local minima [3]. Unlike gate-based quantum computing, QA is designed to find low-energy solutions to problems expressed as Ising models or, equivalently, as Quadratic Unconstrained Binary Optimization (QUBO) problems. This paradigm has been realized in practical hardware via systems like the D-Wave Advantage, which uses thousands of superconducting qubits [4, 5].

Formulating optimization problems as QUBOs is central to harnessing QA effectively. A comprehensive mapping of classical NP-hard problems to QUBO and Ising forms, demonstrating the model's flexibility across domains including graph theory, scheduling, and statistical inference was shown in [6]. Subsequent research extends this work to machine learning, showing how QUBOs can directly encode loss functions and regularization terms for training models [7].

In the context of feature selection (Task 1) and related ML tasks, several studies have explored QA methods using both filter and wrapper approaches. A QUBO framework that encodes feature importance and redundancy was introduced by [8], influencing later work by [9] and [10], who adapted this for recommender systems. Systematic studies and feature selection techniques, expanding on hybrid solver architectures were done by [11] and [5]. Repository-scale applications of QA have also emerged: [9] performed quantum-annealing-based feature selection in a diverse set of classical supervised learning tasks. Feature selection was adapted with QA for recommending content in sparse scenarios, addressing real-world scalability in [10]. Their demonstration of combining relevance and redundancy within the QUBO matrix for domain-specific datasets closely aligns with our methodology.

In the context of instance selection (Task 2), [12] introduced the first Quantum Annealing approach for instance selection problem and indeed proposes the first QUBO formulation for it. It is a straightforward application of cosine similarity between document embeddings and a size constraint encoded into the objective. While their formulation is straightforward, it laid the foundation for subsequent refinements.

In parallel, approaches like E2SC[13] and influence-function-based methods [14, 15, 16] offer algorithms for instance selection in a classical computing paradigm.

With regards to Clustering (Task 3), [17] introduced one of the earliest QUBO formulations for the k-Medoids clustering problem, proposing a binary optimization objective that selects k representative medoids from a dataset without requiring explicit cluster assignments. Their formulation directly inspires our refinement stage, as we adopt their objective structure and constraint encoding to enforce fixed-size cluster selection using quantum annealing. Unlike their purely theoretical framing, however, we embed this QUBO formulation into a full pipeline that combines classical pre-clustering with quantum refinement, tailored to work within real-world hardware limits. Building on this, [18] applied QUBO-based k-Medoids clustering in a document retrieval context for QuantumCLEF 2024. They implemented a hierarchical method that uses simulated annealing and classical clustering for dimensionality reduction before quantum refinement. Their work demonstrates the promise of combining classical preprocessing with quantum optimization for large-scale embeddings, a structure we also adopt. However, our approach differs by systematically comparing multiple classical clustering methods (e.g., k-Medoids, HDBSCAN, GMM) and integrating a principled formulation of the fixed-k constraint using `dimod.generators.combinations`, enabling more consistent enforcement during sampling.

QA-ST was proposed by [19], a quantum annealing-based clustering algorithm that extends simulated annealing using a quantum effect to explore multiple suboptimal solutions. Their results show that quantum annealing can outperform simulated annealing (SA) in exploring global optima across datasets such as MNIST and Reuters. While their work focuses on probabilistic exploration within the clustering assignment space, ours emphasizes post-clustering refinement—using quantum annealing to select diverse, high-quality medoids from a pre-clustered pool under strict constraints, which is critical in information retrieval contexts. A novel perspective is contributed by [20], leveraging all samples returned from a quantum annealer to build calibrated posterior distributions over balanced k-means clusterings. Their probabilistic approach enables uncertainty quantification and ambiguity detection. In contrast, our work prioritizes determinism and fixed-k control, optimizing medoid selection to support retrieval performance rather than exploring ensemble uncertainty. A hybrid clustering method is introduced by [21], combining quantum-inspired optimization with classical updates to handle imbalanced data. Their simulated bifurcation method offers fast discrete optimization with high-quality results, yet focuses on cluster balance in traditional assignments. Our pipeline, by contrast, is structured for downstream document retrieval and focuses on interpretability, medoid diversity, and robust fixed-k constraints.

In summary, prior research lays important groundwork for QUBO-based clustering and hybrid quantum-classical approaches. Our contribution builds directly on these insights, but advances them through (1) a principled, modular pipeline for real-world document clustering; (2) comparative evaluation of multiple classical clustering strategies upstream of quantum refinement; and (3) robust enforcement of exact medoid count using optimized QUBO constraint encodings. Together, these additions bridge the gap between theoretical clustering formulations and practical, retrieval-oriented quantum applications.

3. Methodology

3.1. QUBO Formulation for Quantum Annealing

Quantum annealing is a computational process that uses quantum mechanics to find the best solution to complex optimization problems. It relies on the adiabatic theorem of quantum mechanics, which states that a quantum system initially in the ground state of a known, simple Hamiltonian will remain in the ground state if the system evolves slowly enough and the Hamiltonian is changed gradually [3]. In QA, this principle is used to guide the system from an initial Hamiltonian with a known ground state to a final Hamiltonian that encodes the objective function of an optimization problem. If the evolution follows the conditions of the quantum adiabatic theorem, the system is expected to remain in its ground state, thereby yielding the optimal solution.

To apply quantum annealing to a problem, it must first be formulated as a Quadratic Unconstrained Binary Optimization (QUBO) problem [6]. A QUBO is defined as:

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{Q} \mathbf{x} \tag{1}$$

$$f(\mathbf{x}) = \sum_i Q_{ii} x_i + \sum_{i<j} Q_{ij} x_i x_j \tag{2}$$

where $\mathbf{x} \in \{0,1\}^n$ is a binary vector encoding decisions (e.g., feature, instance, or medoid selection), and $\mathbf{Q}$ is an $n \times n$ matrix representing the cost or similarity structure among variables. Q is the QUBO matrix whose diagonal and off-diagonal entries encode the linear weights and pairwise interactions, respectively. The entries of the QUBO matrix Q can be interpreted in terms of their role in the objective function. The diagonal terms Q_{ii} represent the linear coefficients associated with individual binary variables x_i, and they determine how much each variable contributes to the total cost when it is set to 1. The off-diagonal terms Q_{ij} for $i \neq j$ capture the pairwise interactions between variables x_i and x_j. A negative off-diagonal entry encourages both variables to take the same value (e.g., both 1), while a positive value penalizes such configurations, promoting diversity or mutual exclusion. This structure allows QUBO to naturally encode constraints and preferences between variables, making it suitable for representing complex optimization problems like feature redundancy minimization or balanced clustering. The goal of quantum or classical annealing is to find the binary vector $\mathbf{x}$ that minimizes this objective function f($\mathbf{x}$). This formulation serves as the foundation across all tasks in our pipeline, with task-specific adaptations encoded through the construction of $\mathbf{Q}$. While QUBO problems tend to be "unconstrained", we can add a penalty term to the QUBO formulation that allows the problem to have a soft constraint.

To solve QUBO problems via quantum annealing, we use the D-Wave Advantage_System4.1 quantum processor. This device consists of 5,760 superconducting qubits laid out in a Pegasus P16 topology, which offers enhanced connectivity and embedding flexibility compared to earlier architectures like Chimera [4]. The QUBO problems are submitted through D-Wave's Ocean SDK[22], which handles the necessary problem embedding, chain construction, and solver parameter configuration. The access to D-Wave's quantum annealers was provided to us by the qCLEF organizers through a specialized infrastructure. For comparison, we also evaluate simulated annealing (SA) using D-Wave's classical solver under similar settings. By running both solvers across the same QUBO formulations, we explore the effectiveness, quality, and consistency of quantum annealing versus classical methods in solving ML-driven optimization problems. The specific formulations for each task are discussed in the following sections.

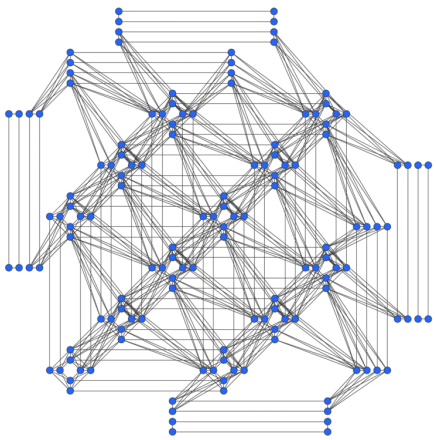


Figure 1: The structure of a Pegasus topology, which defines the connection pattern between qubits in a D-Wave Advantage quantum chip. Blue dots are nodes (qubits) and connecting edges are couplers[23]

3.2. Task 1: Feature Selection

Feature selection is a fundamental preprocessing step in many supervised learning pipelines. The goal is to identify a subset of informative, non-redundant features that improve model generalization and reduce overfitting. We formulate feature selection as a combinatorial optimization problem suitable for quantum annealing by leveraging the framework proposed by [8]. Their approach encodes a balance of *feature importance* and *redundancy* directly into a QUBO matrix, making it amenable to solvers like D-Wave.

In our formulation, the *QUBO matrix* Q is constructed such that:

- The **diagonal entries** Q_{ii} represent **importance scores** of individual features.
- The **off-diagonal entries** Q_{ij} encode **redundancy** between feature pairs.
- A **penalty term** is included to enforce sparsity and encourage the selection of exactly k features. This is formulated as a quadratic penalty on the number of selected features, e.g., $\lambda \left(\sum_i x_i - k\right)^2$. The penalty term also allows us to explicitly control the number of features selected, tuning k based on performance.

This formulation incentivizes selecting features that are individually relevant while penalizing redundancy and constraining the number of selected features via the penalty term.

Importance and Redundancy Measures

For the MQ2007 dataset, we evaluated multiple configurations of Q, combining the following measures:

Importance measures (used for Q_{ii}):

- **Mutual Information (MI)** between feature X and target label Y [24]:

$$\mathrm{MI}(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right)$$

- **Permutation Feature Importance (PFI)**, defined as the change in model error after permuting feature X [25]:

$$\mathrm{PFI}(X) = \mathbb{E}[\mathrm{Error}_{\mathrm{perm}(X)} - \mathrm{Error}_{\mathrm{original}}]$$

Redundancy measures (used for Q_{ij}, $i \neq j$):

- **Conditional Mutual Information (CMI)** between X and Y given Z, estimated between pairs of features conditioned on the target [24]:

$$\mathrm{CMI}(X;Y \mid Z) = \sum_{x,y,z} p(x,y,z) \log \left(\frac{p(x,y \mid z)}{p(x \mid z)p(y \mid z)} \right)$$

- **Conditional Permutation Feature Importance (CPFI)**, which measures the importance of features X and Y when used together [26]:

$$\mathrm{CPFI}(X,Y) = \mathbb{E}[\mathrm{Error}_{\mathrm{perm}(X,Y)} - \mathrm{Error}_{\mathrm{original}}]$$

We experimented with several combinations (e.g., MI+CMI, PFI+CPFI) to populate Q, and selected the combination yielding the best classification accuracy on validation data.

Large-Scale Adaptation for the Istella Dataset

For the *Istella dataset*, which contains significantly more features, we limited our analysis to the MI+CMI combination due to computational constraints. Notably:

- Computing CMI for all feature pairs is computationally expensive. To scale this, we used Python's `multiprocessing.Pool()` to parallelize the computation, reducing runtime considerably.
- The resulting QUBO matrix was too large to be embedded directly onto the D-Wave Advantage system. To address this, we used the `LeapHybridSampler()`, which combines classical and quantum resources to solve large QUBOs that exceed qubit count or connectivity limitations.

This hybrid strategy allowed us to evaluate the viability of QUBO-based feature selection even on larger, real-world datasets.

3.3. Task 2: Instance Selection

The second application deals with instance selection, selecting a subset of instances (i.e. a coreset [16]) of document embeddings with the goal of fine-tuning an LLM on that selected subset, in a subsequent step. Here we only address the general instance selection challenge as the fine-tuning itself was outside of the scope of the competition. As with all other QA problems, instance selection has to be transformed into a QUBO problem first (possibly by incorporating the constraints in the target function) as per equation 3.1. To that end, we used the backbone of the *bcos* algorithm considered in [12] which constructs the diagonal and off-diagonal elements of the Q-matrix. Another aspect that we took from [12] relates to handling of the problem size on the QPU: we batched the dataset in batches of size 80 and processed the data per batch. All the Q matrix entries take that into account - they are calculated on a per-batch basis.

For the off-diagonal entries $Q_{i,j}$, the *bcos* algorithm it considers two cases between each two embeddings for each document pair (doc_i, doc_j): $-cos(doc_j, doc_i)$ if doc_j and doc_i have different labels; $cos(doc_j, doc_i)$ if doc_j and doc_i have the same label. In our work, we kept the off-diagonal entries in same logic as in *bcos* and for the **diagonal** terms $Q_{i,j}$, where $i = j$, we investigated the following extensions:

svc-method Adding a penalty term in the following way: First running a simple (in-sample) support-vector-classifier on all documents within a fold, with the document label as a target and the embeddings as features. Subsequently, by extracting the distance to the fitted support vector of each instance in the fit. Denoting with $\widehat{d}_i^{svc}$ the (estimated) distance to the margin for each instance i, we have for each diagonal entry $Q_{i,i} = \frac{1}{\widehat{d}_i^{svc}+1e-12}$. Lower distances should get higher weight in the Q-matrix as they are more important for the classification. The entries are subsequently normalized before running the QA step. The hyperparamaters of the support vector classifier here are not that important (after some experimentation we settled an *rbf* kernel with a gamma parameter $C = 1.0$ for all experiments), as the goal of the distance metric is to establish a relative ranking between the instances.

instance-deletion Borrowing motivation from Cook's distance as a measure of influence of a sample point, we ran a simple iterative instance deletion model (logistic regression) measuring the decrease of performance when removing each datapoint [15, ch. 31] within a fold. Our goal was to produce a simple heuristic that measures the direct impact of an instance to a classification problem and inspired the choice of the logistic regression as a model that is very fast to compute. The entry for each diagonal element of the Q matrix within a batch b is then simply the value of the influence measure for the effect on the model prediction:

$$Q_{i,i} = \frac{1}{n_b} \sum_{k=1}^{n_b} \left| \widehat{y}_k - \widehat{y}_k^{(-i)} \right| \tag{3}$$

, which is akin to the numerator of Cook's distance, by changing the functional value from squared distance to absolute distance following [ch. 31][15]. More complex versions of such measurement instance influence exist and would be subject to further studies, e.g. ([15], [12])

For our submission, we also included tests with the vanilla *bcos* method. All methods feature an enforced constraint such that the desired level of size reduction is achieved, as in [12].

3.4. Task 3: Clustering

This research focuses on a document clustering and retrieval pipeline that combines classical machine learning techniques with quantum annealing to address the challenges of working with high-dimensional embedding spaces. The core methodology follows a structured two-stage approach:

1. Reduce and summarize the data using classical clustering algorithms (e.g., k-Medoids, HDBSCAN, GMM) to generate candidate medoids.
2. Apply quantum annealing to refine medoid selection using a constrained QUBO formulation.

The pipeline begins by loading high-dimensional document and query embeddings. To support faster clustering and enable more efficient experimentation, dimensionality reduction using Uniform Manifold Approximation and Projection (UMAP) was explored as an optional preprocessing step. UMAP works by modeling local neighborhood relationships in the high-dimensional space as a graph and then optimizing a low-dimensional representation that preserves both local and global structure. When used, this reduction accelerated the initial clustering process and aided in visualizing the overall document distribution, while reproducibility was ensured through consistent random seeds.

In the first stage, a classical clustering algorithm, selected from k-Medoids, HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise), GMM (Gaussian Mixture Model), or a hybrid HDBSCAN-GMM approach, is applied to generate an overcomplete set of candidate medoids. These are representative data points that summarize local structure in the embedding space and serve as a compressed input to the quantum stage. This compression is necessary due to the limited scale of current quantum annealing hardware, which cannot operate over the full embedding space.

Each clustering algorithm introduces different structural assumptions and was evaluated independently to explore how these influence downstream refinement. K-Medoids was used for its emphasis on compact, interpretable clusters, with automatic selection of k via silhouette and Davies-Bouldin Index optimization. HDBSCAN provided a density-based alternative, able to discover clusters of arbitrary shape and automatically discard low-signal regions as noise. GMM framed clustering probabilistically as such, producing soft memberships that captured overlapping semantic regions in the embedding space. The hybrid HDBSCAN-GMM approach layered these strengths by first isolating dense cores with HDBSCAN and then modeling their uncertainty with GMM. While only one algorithm is used in any given run, this flexibility allowed the pipeline to examine how different clustering assumptions affect the quality and diversity of medoid candidates.

The second stage builds on the general QUBO formulation described in Eq. (1), refining the candidate medoids by solving a constrained optimization problem tailored to clustering. The specific formulation we adopt is based on [17], which identifies representative medoids without explicitly clustering the data. To compute pairwise dissimilarities between candidate medoids, we use Welsch's M-estimator, which transforms squared Euclidean distances D_{ij} into robust similarity scores:

$$\Delta_{ij} = 1 - \exp\left(-\frac{1}{2}D_{ij}\right) \tag{4}$$

This formulation, also known as the correntropy loss [27], emphasizes small distances while suppressing the influence of outliers.

The weighted QUBO objective used for medoid refinement is given by:

$$f(\mathbf{x}) = \mathbf{x}^T\left(\gamma\mathbf{1}\mathbf{1}^T - \frac{\alpha}{2}\Delta\right)\mathbf{x} + \mathbf{x}^T\left(\beta\Delta\mathbf{1} - 2\gamma k\mathbf{1}\right) \tag{5}$$

Here, $\mathbf{x} \in \{0,1\}^n$ indicates medoid selection, Δ is defined in Eq. (2), $\mathbf{1}$ is the all-ones vector, and k is the desired number of medoids. Following Eq. (3), we set $\alpha = \frac{1}{k}$ and $\beta = \frac{1}{n}$ to normalize contributions from the dispersion and centrality terms, and use $\gamma = 2$ to prioritize the fixed-k constraint. This formulation directly informs our quantum objective matrix $\mathbf{Q}$ and provides principled control over medoid selection behavior.

The QUBO objective (Eq. 5) encodes both pairwise dissimilarities between medoids and a hard constraint enforcing the selection of exactly k clusters, expressed as $\sum_{i=1}^{n} x_i = k$. This exact constraint is central to the pipeline's design, enabling fixed-k clustering in settings where classical methods often return variable or heuristically chosen cluster counts. We initially experimented with several ways to enforce the fixed-k constraint, including adding a quadratic penalty term $(\sum x_i - k)^2$, post-filtering infeasible samples, and scaling penalty weights. While these worked moderately well with simulated annealing, quantum annealing frequently failed to return exactly k medoids, particularly at small k, due to noise and the relatively weak enforcement of linear or diagonal penalties. The quadratic form, while mathematically equivalent, induces pairwise correlations between all variables, creating a steep energy valley that better resists hardware noise and fluctuations. Motivated by this, we shifted to a more principled approach: we first constructed the clustering loss and then applied the fixed-size constraint using `dimod.generators.combinations`, which implements the same quadratic constraint in a way optimized for quantum hardware [28]. To enforce the fixed-k constraint in practice, we scaled the associated penalty using the maximum energy delta of the clustering term and found that doubling this value consistently stabilized solutions across k and solvers. All quantum and simulated annealing runs used 100 reads per solve. This formulation (Eq. 3) proved to be the most robust across both simulated and quantum annealing settings, offering clean separation between clustering structure and constraint enforcement.

Following refinement, all documents are reassigned to the nearest selected medoid using the original, unreduced embedding space. This separation between reduced-space clustering and full-space evaluation ensures that the final cluster assignments remain faithful to the original data distribution. Cluster quality is measured using the Davies-Bouldin Index (DBI), a metric that balances intra-cluster compactness and inter-cluster separation. To assess retrieval effectiveness, the pipeline matches query embeddings to cluster centroids and ranks documents within each cluster by similarity. Retrieval metrics such as nDCG@10 and relevant document coverage are computed to quantify how well the clusters support downstream information access. Overall, this methodology combines the interpretability and scalability of classical clustering with the constraint-enforcing capabilities of quantum optimization. By decoupling the tasks of structural summarization and hard cluster selection, the pipeline makes principled use of quantum resources where they are most effective, optimizing over a reduced, meaningful subset of the data, while retaining the flexibility to experiment with different clustering assumptions upstream.

4. Results and Discussion

4.1. Task 1: Feature Selection

In this section, we reflect on the results of our experiments across both simulated annealing (SA) and quantum annealing (QA) methods for feature selection. Our primary strategy was to evaluate various combinations of importance and redundancy metrics and to tune the number of selected features, denoted by k, to maximize performance on a held-out validation set. Based on this tuning, we selected the best-performing configurations to submit to the qCLEF leaderboard.

For simulated annealing, we explored a range of k values, from 5 to 40 (out of the total 46 features for the MQ2007 data), and analyzed their corresponding performance using both local evaluation (nDCG@10) and leaderboard scores (Table 1). Among the different configurations, those involving mutual information and conditional mutual information (MI + CMI) showed strong performance on the validation set. We hypothesize that this may be attributed to the model-agnostic, information-theoretic nature of MI and CMI, which allows for more consistent estimation of feature relevance and redundancy.

However, this advantage appears less pronounced on the held-out test set, where configurations based on permutation feature importance (PFI) also performed competitively, particularly when evaluated using LightGBM (LGB) as the underlying model. Notably, LGB-based methods produced the highest nDCG scores in our local experiments. Unfortunately, we were unable to submit LGB-based feature sets to the shared qCLEF evaluation infrastructure due to compatibility issues. The LightGBM package relies on system-level OpenMP support, specifically the `libgomp.so.1` shared library. This library was not provided in our restricted qClef development environment, leading to runtime import errors and preventing the use of LightGBM. Thus, we were limited to using XGBoost (XGB) for official submissions. This constraint may have impacted the final leaderboard performance of otherwise stronger feature selection combinations.

Table 1
nDCG@10 scores across different feature selection strategies using Simulated Annealing (SA). Each strategy corresponds to a combination of importance measures (e.g., MI, PFI) used on the diagonal of the QUBO matrix and redundancy measures (e.g., CMI, CPFI) used on the off-diagonal.

k	**MI + CMI**	**PFI + CMI (lgb)**	**PFI + CPFI (lgb)**	**PFI + CPFI (xgb)**	**PFI + CMI (xgb)**
5	0.3817	0.2002	0.1898	0.3811	0.3671
10	0.5232	0.6509	0.2474	0.1931	0.4865
15	0.5897 (0.4485)‡	0.3431	0.4531	0.5215	0.1847
20	0.4670	0.4142	0.5419	0.5745 (0.4318)‡	0.3255
25	0.5859 (0.4510)‡	0.5123	0.4655	0.5396	0.5845 (0.4500)‡
30	0.4003	0.4675	0.6873	0.5646	0.5679 (0.4523)‡
35	0.3970	0.6211	0.4312	0.5398	0.5688
40	0.3866	0.3866	0.5781	–	–

Each cell shows the validation nDCG@10 score. These were calculated on the validation set.
The baseline nDCG@10 score including all 46 features is 0.4473.
‡ Configuration submitted to the qCLEF leaderboard. Values in parentheses are the official CLEF leaderboard scores calculated on the held-out test set.
MI: Mutual Information; PFI: Permutation Feature Importance; CMI: Conditional Mutual Information; CPFI: Conditional Permutation Feature Importance.
PFI and CPFI were computed using LightGBM (lgb) and XGBoost (xgb) based importance scores.
All results shown are from local validation runs. While LightGBM (LGB) models often outperformed XGBoost (XGB) in our local evaluations, we were unable to submit LGB-based models to the shared qCLEF evaluation infrastructure due to compatibility issues. As a result, only XGB-based feature sets were submitted for final leaderboard scoring.
"–" indicates configurations that were not evaluated due to resource constraints.

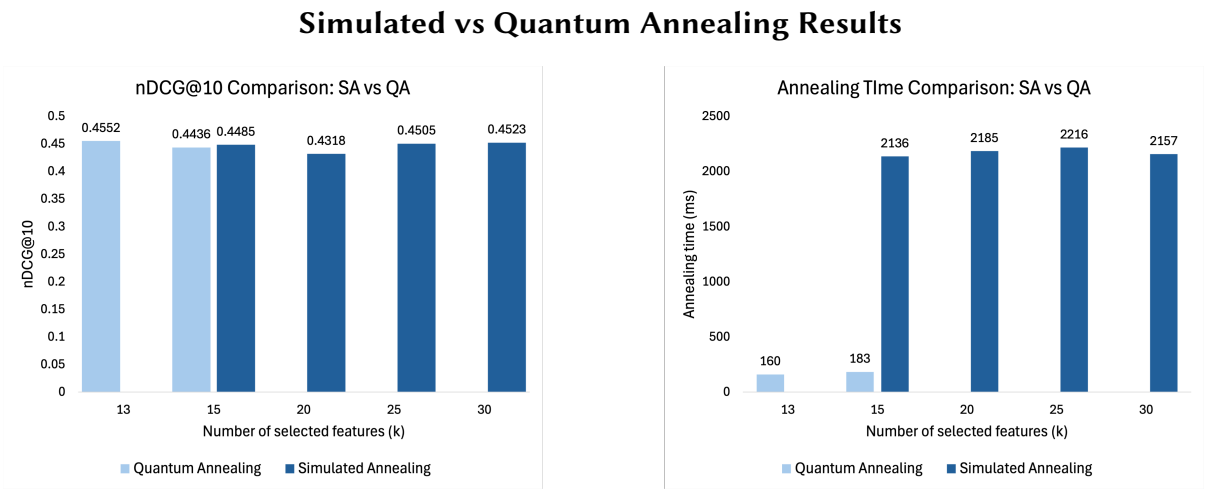
For quantum annealing, despite our interest in conducting experiments for more configurations, we were constrained by limitations in the quantum infrastructure. Specifically, the time and resource availability for the D-Wave quantum annealer limited the breadth of our QA experiments. As a result, we were only able to submit two QA-based runs, both derived from the same codebase and configuration (Table 2). Interestingly, these two QA submissions resulted in different outcomes: one returned a feature subset of size 13 with an nDCG of 0.4552, while the other selected 15 features and achieved an nDCG of 0.4436. This divergence is notable because the code and QUBO formulation were identical in both cases. We attribute this variance to the inherent randomness and probabilistic nature of the quantum annealing process, where solution quality can fluctuate between runs due to quantum noise, minor differences in embedding, or hardware-level stochasticity.

Table 2
CLEF leaderboard nDCG@10 scores for Quantum Annealing (QA)-based feature selection. All configurations used the D-Wave Advantage_system quantum annealer and Mutual Information (MI) as the importance measure on the diagonal.

Method	k	Leaderboard nDCG@10
QA (MI-CMI)	15	0.4436
QA (MI-CMI)	13	0.4552

These configurations were submitted directly to the CLEF leaderboard without local validation.
The QA runs were executed using D-Wave's Advantage_system with 5760 qubits and Pegasus topology.

An interesting result is that the QA submission with just 13 features achieved the highest nDCG score among all our submissions, and notably, it also had the fewest selected features among all leaderboard entries. While the top leaderboard entry achieved an nDCG of 0.4580 using 21 features, our QA submission reached a comparable score of 0.4552 with only 13 features. This makes it arguably the most efficient feature subset in terms of predictive performance per feature used.



(a) nDCG@10 comparison across SA and QA.

(b) Annealing time (ms) for SA and QA.

Figure 2: Comparison of Simulated Annealing (SA) and Quantum Annealing (QA) on feature selection task. Values in the chart only reflect the submissions made to the qCLEF leaderboard and official results.

Furthermore, while SA and QA achieved comparable nDCG scores across the board, the computational effort differed significantly (Figure 2). Our analysis shows that QA completed the optimization process in approximately one-tenth the time required by SA for similar configurations (see 2b). This suggests that quantum annealing may offer a more efficient route to high-quality solutions, especially in time-sensitive or resource-constrained environments. Overall, these findings underscore the potential of quantum annealing not just as a novelty, but as a competitive alternative to classical metaheuristics like simulated annealing for tasks like feature selection in machine learning pipelines.

4.2. Task 2: Instance Selection

The way to properly evaluate an instance selection routine in the context of a QA routine is based on a tripod of criteria, as noted in [12]: size reduction, performance and total inference time. Naturally, there can be a situation where no method dominates the others in all three aspects. We decided to focus on F1 score, as there was no guideline regarding the reduction size - all of our submissions were at a size of a targeted reduction size of 25% - i.e. 75% of the instances were targeted to be kept. Table 3 shows the competition results achieved by our team. We had managed to perform one quantum run which we kept - using the *bcos* method, which does not achieve exact 25% reduction due the inherent

randomness in the quantum annealing procedure, but all the simulated annealing runs are at exacty 25% reduction. As evident by the standard deviation numbers (in brackets), while we nominally top the leaderboard for a fixed reduction size, the differences to the baseline and to the other teams' submissions are not statistically significant for the *yelp* dataset. For the *vader* dataset all teams perform worse than the baseline, which remains puzzling as our own analysis indicates a much higher performance than indicated by the leaderboard.

Table 3
DS@GT and Baseline results with overall rankings (rank is against *all* submissions for each dataset)

Rank	Team	Name	F1 Score	Size	Time	Type
		Yelp Dataset				
1	DS@GT qClef	Yelp_SA_qclef_bcos_075	99.5(0.2)	0.25	1548.5(2.8)	S
2	DS@GT qClef	Yelp_QA_qclef_bcos	99.4(0.2)	0.274	1500(54.7)	Q
2	Baseline	BASELINE_ALL	99.4(0.1)	–	2027.1(1.1)	–
3	DS@GT qClef	Yelp_SA_qclef_it_del_075	99.3(0.3)	0.25	1549.2(1.5)	S
3	DS@GT qClef	Yelp_SA_qclef_svc_075	99.3(0.4)	0.25	1550.5(2.6)	S
		Vader Dataset				
1	Baseline	BASELINE_ALL	88.9(0.8)	–	1997.3(5.7)	–
2	DS@GT qClef	Vader_SA_qclef_combined_075	65.9(4.7)	0.25	1529.4(3)	S
3	DS@GT qClef	Vader_SA_qclef_it_del_075	65.6(3)	0.25	1529.5(2.3)	S
4	DS@GT qClef	Vader_SA_qclef_svc_075	65.4(7.1)	0.25	1529(2.4)	S
7	DS@GT qClef	Vader_QA_qclef_bcos	62.6(7.5)	0.283	1493.3(83)	Q
8	DS@GT qClef	Vader_SA_qclef_bcos_075	62.5(10.4)	0.25	1528.6(2.2)	S

One insight that can be inferred is that the datasets are simply too trivial for this task. [12] also analyzed these datasets (albeit not in the context of LLM, but of BERT models as a downstream task) and recorded performance of the reduced dataset (by 25%) at the level of the the full dataset's performance. Another argument that supports this insight can is illustrated by the analysis depicted in figure 3. In it, we performed (on the test sets across the different folds provided by the organizer) simple logistic regression model as a substitute of the LLM fine-tuning step, which was not accessible to us as competition participants. The average (across test folds) F1 score is shown for all methods, including a simple random sample method, which drops 25% of the observations within a training fold randomly. They remain fairly stable even for high levels of reduction (10% to 60%), as evidenced also by other teams' submissions on the leaderboard who submitted runs with higher reduction level. We also conducted experiments with fine-tuning BERT models and results were comparable - at the 25% reduction level, there was not a significant difference between a random sampling method and all other methods.

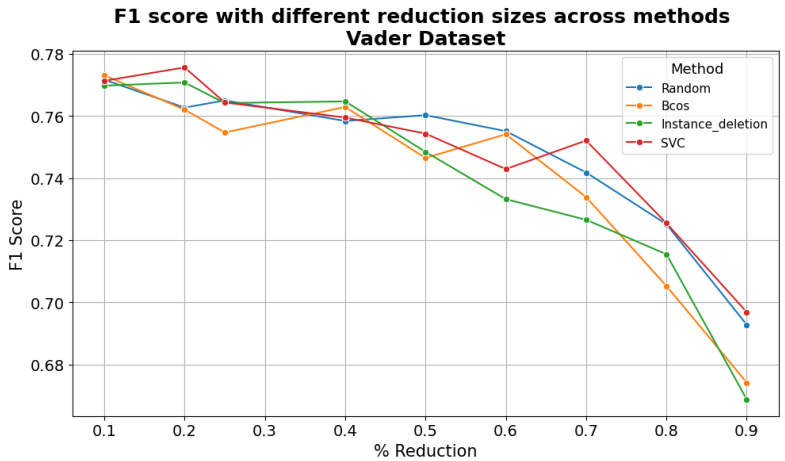


Figure 3: Vader dataset: Average Performance (F1 Score) across test folds of a logistic regression model ran with different instance selection variants and reduction sizes: random sampling, distance to support-vector based method (svc) as well as instance deletion method.

Overall, the (simple) heuristics presented here do not exhibit a significant difference across different reduction levels - neither on the LLM evaluations done on the leaderboard, nor in the simple logistic regression evaluations shown in Figure 3. Likely, this is due to the size of the dataset and the (low) difficulty of the classification task. A more difficult benchmark dataset could help study these differences in higher detail.

Nonetheless, the SVC-method does show certain promise as being the best performing at higher reduction levels and could be a good starting point for further research.

4.3. Task 3: Clustering

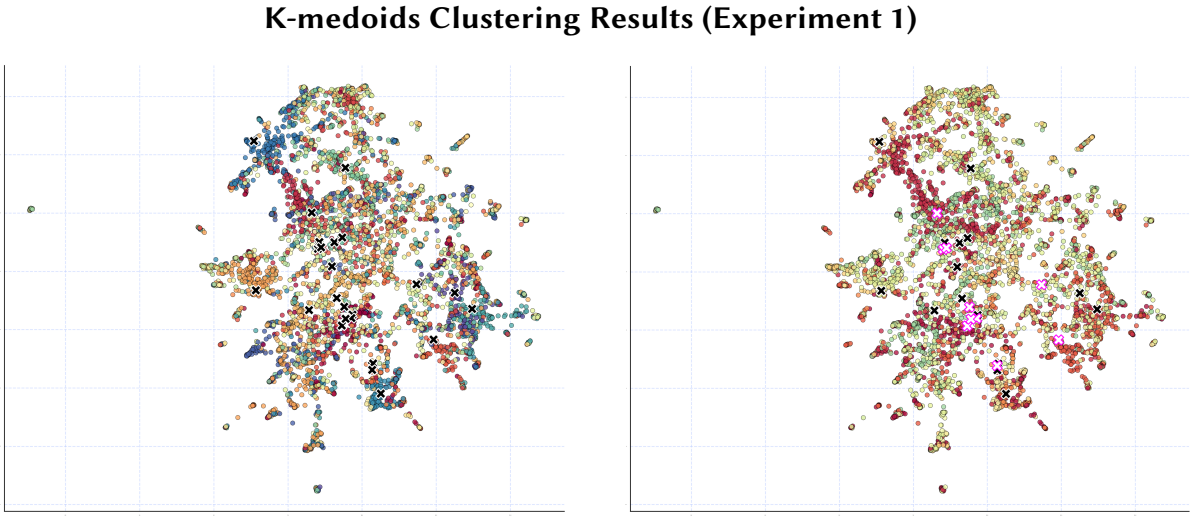
Table 4
Clustering performance across different classical clustering methods and configurations (Runs 1–18), all followed by the same quantum-constrained refinement stage. Baselines are fully classical with no quantum post-processing.

	Classical Medoid Generation	k	UMAP-Reduced	DBI	DBI (leaderboard)	nDCG	nDCG (leaderboard)
1	k-Medoids	10	N	7.4776	7.4776	0.48	0.58
2	k-Medoids*	10	Y	6.2839	4.4706	0.50	0.0172
3	k-Medoids	25	N	6.2554	–	0.48	–
4	k-Medoids	25	Y	5.1365	–	0.44	–
5	k-Medoids	50	N	4.6063	–	0.36	–
6	k-Medoids	50	Y	3.7141	–	0.32	–
7	HDBSCAN	10	N	3.1862	–	0.52	–
8	HDBSCAN	10	Y	6.1444	–	0.56	–
9	HDBSCAN	25	N	5.2627	–	0.46	–
10	HDBSCAN	25	Y	5.0119	–	0.46	–
11	HDBSCAN	50	N	4.3931	–	0.50	–
12	HDBSCAN*	50	Y	3.9469	3.4217	0.44	0.0064
13	GMM	10	Y	5.9105	–	0.60	–
14	GMM	25	Y	5.2283	–	0.36	–
15	GMM	50	Y	4.5136	–	0.38	–
16	HDBSCAN-GMM	10	Y	6.2197	–	0.48	–
17	HDBSCAN-GMM	25	Y	5.1848	–	0.54	–
18	HDBSCAN-GMM	50	Y	3.9478	–	0.50	–
–	Baseline	10	–	–	7.9892	–	0.5509
–	Baseline	25	–	–	6.1201	–	0.5284
–	Baseline	50	–	–	5.3679	–	0.4656

DBI: Davies-Bouldin Index (lower is better); nDCG: Normalized Discounted Cumulative Gain (higher is better).
Internal scores are computed on the training set; leaderboard scores reflect performance on the held-out test set.
Experiments 1, 2, and 12 reflect submitted results with experiment 1 achieving top score for the task.
*Dimensionality-reduced centroids were included in the final submission, leading to evaluation errors.

We evaluated a range of clustering configurations to explore how different classical methods and quantum refinement strategies impact retrieval effectiveness. Table 4 summarizes the results from both submitted and exploratory experiments. Among the submitted runs, Experiment 1 (k-Medoids, $k = 10$, no UMAP) achieved the highest performance, with a leaderboard nDCG of 0.58 and an internal validation score of 0.48, outperforming all baselines. This strong result can be attributed to the simplicity

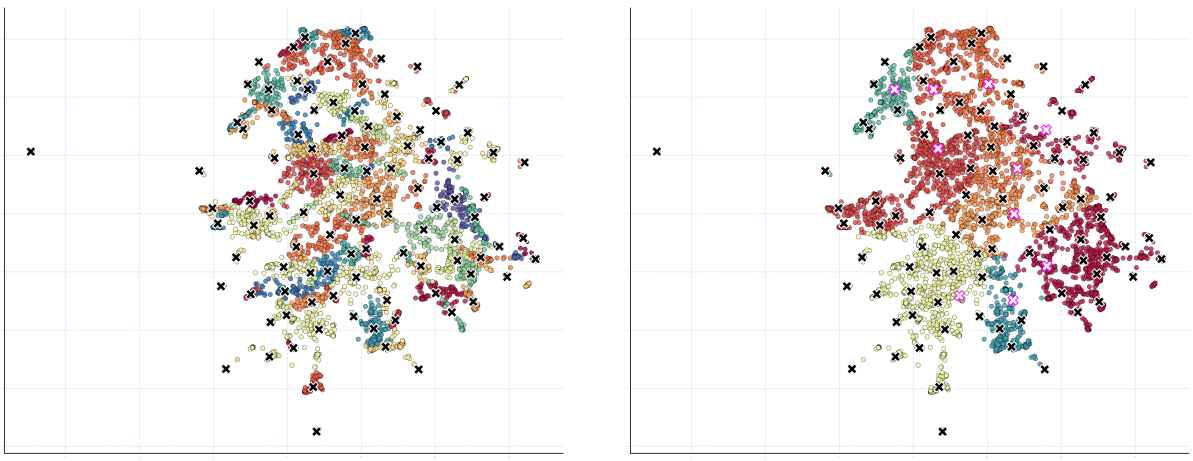
and structure-preserving nature of the two-step k-Medoids pipeline, which maintained the original semantic geometry of the embedding space and yielded consistently strong retrieval performance.



(a) Initial clustering (b) Quantum-refined clustering

Figure 4: Experiment 1 (k-Medoids, $k = 10$) clustering visualization using 2D UMAP projection. Left: initial clustering with centroids (black), right: quantum-refined medoid selection with final centroids (pink).

Among all experiments, however, experiment 13 (GMM, k=10) achieved the highest nDCG (0.60) on training data, followed closely by experiment 17 (HDBSCAN-GMM, k=25) with 0.54, and Experiment 7 (HDBSCAN, k=10, no UMAP) with 0.52 and the lowest DBI overall (3.19). Experiment 13's strong performance likely stemmed from GMM's probabilistic flexibility at low k, which captured nuanced topical overlap and yielded the best retrieval quality. Experiment 17 benefited from HDBSCAN's structure-aware initialization, followed by GMM fitting. This hybrid approach, especially at k=25, struck a strong balance between granularity and semantic coherence. In contrast, experiment 7 benefited from density-based clustering (HDBSCAN) applied directly to the high-dimensional space. At k=10, it effectively discovered dense semantic regions, while the quantum refinement stage helped consolidate them into meaningful, noise-filtered clusters.



(a) Initial clustering (b) Quantum-refined clustering

Figure 5: Experiment 13 (GMM, $k = 10$) clustering visualization using 2D UMAP projection. Left: initial clustering with centroids (black), right: quantum-refined medoid selection with final centroids (pink).

The results reveal consistent and interpretable trends across different clustering configurations, particularly with respect to the number of clusters (k), the use of dimensionality reduction, and the behavior of classical clustering methods prior to quantum refinement. Across all methods, increasing k generally led to improved Davies-Bouldin Index (DBI) scores, indicating tighter and more distinct clusters. This was most evident in the k-Medoids experiments, where DBI steadily decreased from 7.48 at k=10 to 3.71 at k=50, reflecting improved intra-cluster compactness and inter-cluster separation. However, while increasing k improved DBI, retrieval quality often peaked at lower k values using soft or structure-aware clustering methods. Baseline retrieval scores followed a similar pattern, dropping from 0.55 at k=10 to 0.47 at k=50, further emphasizing that expressiveness, not just compactness, plays a key role in modeling topical overlap in retrieval settings.

Dimensionality reduction using UMAP was explored as an optional preprocessing step to accelerate clustering and support visualization. While UMAP occasionally led to lower DBI scores as seen in experiments 7 and 8, its impact was not uniformly positive. In many cases, UMAP had little effect on DBI or even slightly worsened it. Moreover, improvements in geometric compactness did not consistently translate into better retrieval performance. In some configurations, especially at lower k, applying UMAP prior to clustering led to lower nDCG values, suggesting that key structural cues for retrieval may be lost in the projection to a reduced space. All GMM-based methods were run with UMAP applied due to their computational cost in high-dimensional space; non-reduced probabilistic clustering was excluded for tractability reasons, though it remains an area for future exploration.

Experiments 2 and 12, which applied UMAP before clustering, mistakenly submitted dimensionally reduced centroids to the leaderboard evaluation. Because retrieval metrics were computed using full-dimensional query embeddings, this mismatch resulted in invalid similarity calculations and artificially low leaderboard scores, especially for nDCG. These values should not be interpreted as indicators of poor clustering quality and instead reflect a representation mismatch during evaluation.

Together, these results validate the two-stage pipeline's strategy of first generating an overcomplete and structurally diverse set of medoid candidates through classical clustering and then refining them using quantum-constrained optimization. The consistent improvements in DBI with higher k and the generally reliable performance of classical methods set a strong foundation for the second-stage quantum refinement, which enforces fixed-k constraints in a principled way.

5. Future Work

5.1. Task 1: Feature Selection

While our study focused on QUBO formulations built from combinations of mutual information, conditional mutual information, and permutation-based importance scores, there remain several promising directions for future exploration. We intend to extend our methodology to incorporate additional feature importance techniques inspired by classical literature. In particular, methods such as *Functional ANOVA* (fANOVA) [29] and *Leave-One-Feature-Out* (LOFO) importance [30] offer intuitive measures of a feature's marginal and conditional relevance within a model context. These could potentially be adapted into the QUBO framework by mapping importance scores to diagonal entries and interactions (e.g., joint relevance or redundancy) to off-diagonal terms. Another promising candidate is the *Relief* family of algorithms [31], which estimate feature relevance based on how well feature values distinguish between near instances of different classes. Since Relief naturally accounts for both relevance and redundancy, it may be especially well-suited for QUBO-based optimization.

5.2. Task 2: Instance Selection

In the context of Task 2, it is important to note that all of the aforementioned computations were done on a per-batch basis. Thus every batch would have a most influential datapoint (with either svc or instance deletion based method) calculated only in relation to the other datapoints in the batch. We kept this as the same batching logic is applied to the off-diagonal terms. Two documents which are very

highly similar (as measured by cosine similarity) to each other (and have either positive or negative label) can thus end up being in two different batches and their cosine similarity will be never taken into account. While batching is necessary as the QPU cannot fit all documents at once, this poses a challenge as the application of the “penalties" and “rewards" for hard instances are not applied on a global level. This is much easier to achieve at least for the diagonal elements however, as the influence points can be easily calculated only once and can be handled independently on the batch (e.g. an svc can be fit using all the training data and the distance from each instance to the support vector can be calculated for each instance, before the batching).

In addition, for the diagonal elements, more complex instance influence function methodology as in [14],[15], [16] can be applied either in combination with, or instead of the simple heuristics presented here.

5.3. Task 3: Clustering

In future work, we would like to evaluate how nDCG scores change when using the quantum processing unit. While we were unable to submit our most promising quantum experiments due to hardware and timing constraints, we remain curious how they would have scored under the competition’s official retrieval metrics on test data. Further future extensions could explore non-reduced probabilistic clustering to assess GMM performance in full embedding space. Additionally, incorporating probabilistic or fuzzy refinement in the quantum stage that may better capture semantic overlap in multi-topic documents.

6. Conclusion

In this work, we investigated the use of quantum annealing (QA) and simulated annealing (SA) to solve key machine learning optimization tasks - feature selection, instance selection, and clustering - by formulating them as QUBO problems. Across all tasks, we developed principled mappings that leveraged both classical and quantum resources effectively.

In Task 1 (feature selection), we explored multiple QUBO formulations that combined different feature importance and redundancy measures. Specifically, we tested combinations of MI, CMI, PFI, CPFI to construct the Q matrix. We evaluated these QUBOs using both simulated annealing and quantum annealing, and found that quantum annealing achieved comparable effectiveness to simulated annealing while requiring significantly less computational effort.

In Task 2 (instance selection), we extended the BCOS algorithm and introduced two new QUBO-based scoring mechanisms derived from SVM margins and instance deletion influence. Despite the lack of statisticaly significant differences between the methods at a reduction level of 25%, these approaches showed promising results even at increased levels of instance reduction and can serve as a basis for further research on more difficult datasets.

Task 3 (clustering) showcased the versatility of hybrid pipelines, combining classical clustering algorithms with quantum-constrained refinement. While the best overall clustering performance was achieved classically, our experiments confirmed that QUBO-based refinement enhances cluster diversity and compactness, particularly in document retrieval tasks.

Across tasks, we found that QA often matched or exceeded the performance of SA in less time, highlighting its potential for more efficient combinatorial optimization. While access to quantum annealers and scale remain ongoing challenges, our findings support the growing viability of quantum annealing as a practical tool in real-world ML pipelines.

Acknowledgments

We thank the DS@GT CLEF team for providing valuable comments and suggestions. We would also like to thank Ayah Zaheraldeen and Jiangqin Ma for their input and support throughout the project. This research was supported in part through research cyberinfrastructure resources and services provided by the Partnership for an Advanced Computing Environment (PACE) at the Georgia Institute of Technology, Atlanta, Georgia, USA.

Declaration on Generative AI

During the preparation of this work, the authors used OpenAI-GPT-4o: Grammar and spelling check. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] A. Pasin, M. F. Dacrema, W. Cuhna, M. A. Gonçalves, P. Cremonesi, N. Ferro, Quantumclef 2025: Overview of the second quantum computing challenge for information retrieval and recommender systems at CLEF, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), Working Notes of CLEF 2025 - Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, 2025.

[2] A. Pasin, M. F. Dacrema, W. Cuhna, M. A. Gonçalves, P. Cremonesi, N. Ferro, Overview of quantumclef 2025: The second quantum computing challenge for information retrieval and recommender systems at CLEF, in: J. Carrillo-de-Albornoz, J. Gonzalo, L. Plaza, A. G. S. de Herrera, J. Mothe, F. Piroi, P. Rosso, D. Spina, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Sixteenth International Conference of the CLEF Association (CLEF 2025), Lecture Notes in Computer Science, 2025.

[3] E. Farhi, J. Goldstone, S. Gutmann, M. Sipser, Quantum computation by adiabatic evolution, arXiv preprint quant-ph/0001106 (2000).

[4] K. Boothby, P. Bunyk, J. Raymond, A. Roy, Next-generation topology of d-wave quantum processors, arXiv preprint arXiv:2003.00133 (2020).

[5] L. P. Yulianti, K. Surendro, Implementation of quantum annealing: A systematic review, IEEE Transactions on Emerging Topics in Computing 11 (2023) 150–162.

[6] A. Lucas, Ising formulations of many np problems, Frontiers in Physics 2 (2014) 5.

[7] P. Date, D. Arthur, L. Pusey-Nazzaro, Qubo formulations for training machine learning models, Quantum Computing Applications 1 (2023) 100–112.

[8] S. Mücke, R. Heese, S. Müller, M. Wolter, N. Piatkowski, Feature selection on quantum computers, Quantum Machine Intelligence 5 (2023) 11. URL: https://doi.org/10.1007/s42484-023-00099-z. doi:10.1007/s42484-023-00099-z.

[9] D. Pranjić, B. C. Mummaneni, C. Tutschku, Quantum annealing based feature selection in machine learning, Quantum Machine Learning 2 (2023) 11–19.

[10] R. Nembrini, M. F. Dacrema, P. Cremonesi, Feature selection for recommender systems with quantum computing, Journal of Computing Frontiers 10 (2024) 45–57.

[11] N. Borle, N. Zecevic, et al., Feature selection with quantum annealing for interpretable and robust machine learning, Quantum Machine Intelligence 5 (2023) 1–15.

[12] A. Pasin, W. Cunha, M. A. Gonçalves, N. Ferro, A quantum annealing instance selection approach for efficient and effective transformer fine-tuning, in: Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval, 2024, pp. 205–214.

[13] W. Cunha, C. França, G. Fonseca, L. Rocha, M. A. Gonçalves, An effective, efficient, and scalable confidence-based instance selection framework for transformer-based text classification, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023, pp. 665–674.

[14] P. W. Koh, P. Liang, Understanding black-box predictions via influence functions, in: International conference on machine learning, PMLR, 2017, pp. 1885–1894.
[15] C. Molnar, Interpretable Machine Learning, 3 ed., 2025. URL: https://christophm.github.io/interpretable-ml-book.
[16] A. S. Joaquin, B. Wang, Z. Liu, N. Asher, B. Lim, P. Muller, N. F. Chen, In2core: Leveraging influence functions for coreset selection in instruction finetuning of large language models, arXiv preprint arXiv:2408.03560 (2024).
[17] C. Bauckhage, N. Piatkowski, R. Sifa, D. Hecker, S. Wrobel, A qubo formulation of the k-medoids problem., in: LWDA, 2019, pp. 54–63.
[18] W. Alvarez-Giron, J. Téllez-Torres, J. Tovar-Cortes, H. Gómez-Adorno, Team qiimas on task 2 - clustering: Quantum annealing for k-medoids optimization, in: Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, Grenoble, France, 2024. URL: https://bitbucket.org/eval-labs/qc24-qiimas/src/main/, cEUR Workshop Proceedings, ISSN 1613-0073.
[19] K. Kurihara, S. Tanaka, S. Miyashita, Quantum annealing for clustering, in: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI), AUAI Press, 2009, pp. 317–324.
[20] J.-N. Zaech, M. Danelljan, T. Birdal, L. Van Gool, Probabilistic sampling of balanced k-means using adiabatic quantum computing, arXiv preprint arXiv:2310.12153 (2023). URL: https://arxiv.org/abs/2310.12153.
[21] N. Matsumoto, Y. Hamakawa, K. Tatsumura, K. Kudo, Distance-based clustering using qubo formulations, Scientific Reports 12 (2022) 2669. URL: https://doi.org/10.1038/s41598-022-06559-z. doi:10.1038/s41598-022-06559-z.
[22] D.-W. S. Inc., Ocean software documentation, 2023. URL: https://docs.ocean.dwavesys.com/.
[23] T. Morstyn, Annealing-based quantum computing for combinatorial optimal power flow, IEEE Transactions on Smart Grid PP (2022) 1–1. doi:10.1109/TSG.2022.3200590.
[24] T. M. Cover, J. A. Thomas, Elements of Information Theory, 2nd ed., Wiley-Interscience, 2006.
[25] L. Breiman, Random forests, Machine Learning 45 (2001) 5–32.
[26] D. Debeer, C. Strobl, Conditional permutation importance revisited, BMC Bioinformatics 21 (2020) 1–19.
[27] W. Liu, P. P. Pokharel, J. C. Principe, Correntropy: Properties and applications in non-gaussian signal processing, IEEE Transactions on Signal Processing 55 (2007) 5286–5298. doi:10.1109/TSP.2007.898255.
[28] J. Pasvolsky, D.-W. S. Inc., dimod.generators.combinations — constraint generator for fixed-k selection, 2019. URL: https://github.com/dwavesystems/dimod/blob/main/dimod/generators/constraints.py, accessed: 2025-06-12.
[29] F. Hutter, H. H. Hoos, K. Leyton-Brown, Efficient functional anova: Insights into high-dimensional model performance, in: Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI), 2014.
[30] A. Abid, A. Kamel, J. Zou, Lofo importance: Leave one feature out based feature importance score, https://github.com/aerdem4/lofo-importance, 2020.
[31] K. Kira, L. A. Rendell, The feature selection problem: Traditional methods and a new algorithm, AAAI (1992) 129–134.