International Journal of Engineering Trends and Technology
ISSN: 2231-5381/ https://doi.org/10.14445/22315381/IJETT-V74I5P121

Volume 74 Issue 5, 312-328, May 2026

Research Article

Novel Optimization Approach for Weighted Metric Evaluation in Question Answering Systems Using Genetic Algorithm and Grey Wolf Optimizer

Priyanka K[1], Toshima Jaiswal[2], Nandhini Kumaresh[3], Jayapriya J[4], Vinay M[5]

[1,2,4,5]*Department of Computer Science, CHRIST University, Bangalore, India.*
[3]*Department of Computer Science, Central University of Tamil Nadu, Tiruvarur.*

[1]*Corresponding Author : priyankakadirvelv@gmail.com*

Received: 14 January 2026 Revised: 06 March 2026 Accepted: 12 March 2026 Published: 30 May 2026

Abstract - *Automated Question Answering(QA) systems are essential building blocks of modern Natural Language Processing, powering a range of informative tasks like virtual assistants, customer support bots, learning tutorial systems, and search engines. With the increasing usage of QA systems, it is also important that evaluation is carried out in a precise and comprehensive manner. The Exact Match and F1-score metrics are mainly focused on word-level similarities, without considering semantic understanding, contextual consistency, and logical consistency. However, it is found that existing QA evaluation schemes are based on fixed metrics or combinations of metrics, which limits their flexibility across different evaluation scenarios and alignment with human judgments. The relative importance of these features could be different with regard to the particular question-answering task or domain. To mitigate such limitations, this paper presents a new, task-adaptive evaluation protocol that blends five heterogeneous and complementary scoring metrics: BERTScore, BLEU, Entailment Score, Normalized Perplexity, and a Contrastive Penalty. Acknowledging that different QA tasks may place different priorities on answer quality, this approach learns optimal weight distributions for each metric component instead of fixed weights. The contribution of this work is the application of two bio-inspired optimization algorithms for making optimal selections of such weights: Genetic Algorithm, which is explicitly used to facilitate better management of human-annotated answer quality with emphasis on contrastive error penalty, and Grey Wolf Optimizer, which optimizes a composite loss function that best balances all the metric components with lower computational overhead. The work also explores a hybrid view by studying and comparing individual strengths of both optimization approaches under a typical experimental setup. Experiments conducted on a curated subset of the SQuAD v2.0 dataset, augmented with contrastive examples to simulate real-world vagueness, demonstrate that both approaches perform better than traditional static metrics in agreement with human judgments. Genetic Algorithm is contrast-sensitive, while Grey Wolf Optimizer is semantically coherent and computationally efficient. These approaches together provide a general, adaptive framework of comprehensive QA evaluation, which could be adapted into various application scenarios.*

Keywords - *Genetic Algorithm (GA), Grey Wolf Optimizer (GWO), BERTScore, BLEU, Entailment Score, Normalized Perplexity, Contrastive Penalty.*

1. Introduction

Question Answering (QA) is a fundamental Natural Language Processing (NLP) task that tries to come up with accurate and contextually important responses to queries that can be worded in natural language. QA systems are now considered part of modern information systems and are used to support the applications of search engines, virtual assistants, customer support services, education tools, and medical information systems. The systems in such applications should retrieve correct answers, but they should also be able to handle the context, ambiguity, and when no adequate evidence to give an answer is available. The latest developments in transformer-based and deep learning have enhanced the functionality of QA systems to a great extent. Nowadays, models like BERT, RoBERTa, and ELECTRA offer a chance to understand language in a deep context based on receiving bidirectional representations of large-scale text corpora. More challenging contexts, like context-dependent reasoning and unanswerable questions, have also been introduced in benchmark datasets like SQuAD v2.0, which has accelerated progress. With these developments, QA models have been able to perform well both on extractive and generative tasks. Nevertheless, in spite of these improvements in the development of models, the assessment of QA systems is a major problem. Conventional measures of evaluation, like Exact Match (EM) and F1-score, are based on lexical overlap

between reference answers and predicted answers. These measures are computationally effective and widely used, albeit they generally only focus on superficial word matching and, in many cases, do not reflect deeper semantic meaning, logical consistency, and contextual relevance. The consequence of this is that semantically correct answers with different wordings to the reference answer can be low scored, whereas semantically shallow answers, which are similar to the reference answer, can be scored incorrectly. Past research has indicated that these metrics often do not align with human judgments.

To address these limitations, embedding-based evaluation measures like BERTScore and BLEURT have been suggested to measure semantic similarity between generated and reference responses. Although such approaches enhance semantic sensitivity, they remain mostly independent measures of evaluation, and they usually use a set of fixed weighting plans.

Practically, the quality of answers is determined by a combination of several complementary features, such as semantic accuracy, linguistic fluency, logical conformity, and the capacity to notice wrongful and misleading answers. Therefore, the use of one evaluation metric is not enough to measure the multiple facets of answer quality that are involved in QA systems.

These limitations highlight a significant research gap: current QA assessment models do not have any mechanisms of adaptation, which will allow them to dynamically incorporate a variety of complementary evaluation measures and be in close correlation with human judgement. Despite the fact that composite evaluation methods have been studied in certain settings, very little research has examined optimization-based methods of learning optimal sets of evaluation measures for different QA tasks and datasets.

In domain-specific QA systems, the requirement for strong evaluation mechanisms is ever more urgent. In technical work like civil engineering, quality assurance systems can help engineers access information in structural design manuals, building codes, technical specifications, and safety regulations. Such areas require precision in the interpretation of semantic meaning, logical truth, and relevance in context to make sound decisions. The standardized measures of evaluation that are mostly based on similarity between lexical information are thus not adequate in terms of the quality of the answers given in such high-stakes settings.

Adaptive optimization techniques offer an opportunity to overcome this challenge. Metaheuristic algorithms are capable of dynamically learning the best combination of metrics through exploration of complex search spaces and adapting the metrics used to evaluate performance on empirical data. Particle Swarm Optimization, Ant Colony Optimization, and Differential Evolution are some of the techniques that have been extensively applied in the machine learning domain to tune parameters and weight features. Nevertheless, there is no research on the application of hybrid or comparative metaheuristic optimization techniques to adaptive QA evaluation. To fill this gap, this paper suggests a dynamically and task-adaptive evaluation framework, which incorporates five mutually complementary measures of divergent aspects of answer quality: BERTScore of semantic similarity, BLEU of lexical overlap, Entailment Score of logical consistency, Normalized Perplexity of linguistic fluency, and a Contrastive Penalty of discriminative assessment. The proposed framework does not impose fixed weights on these metrics, but rather, learns the optimal weight distributions with the help of nature-inspired optimization.

Precisely, two bio-inspired optimization algorithms, namely Genetic Algorithm (GA) and Grey Wolf Optimizer (GWO), are used to optimize the contribution of any of the metrics in the composite evaluation score. The Genetic Algorithm optimizes the correlation of candidate weight vectors by applying selection, crossover, and mutation to optimize the correlation with the answer quality as rated by humans. The Grey Wolf Optimizer, on the other hand, emulates the hierarchical structure and hunting patterns of grey wolves to conduct an exhaustive search of the search space and find high-quality weight parameters with reduced computing cost. Using the comparison of these complementary optimization strategies, the proposed framework explores the effects of various metaheuristic methodologies on metric weighting and performance of evaluation.

The major findings of the research are as follows:

- A multi-dimensional evaluation framework of QA based on the combination of semantic, lexical, logical, and fluency-based measurements to offer a more detailed evaluation of the quality of answers.
- An adaptive metric-weight optimization method based on nature-inspired algorithms, which permits the effects of metric weights to be dynamically adjusted to various QA tasks and datasets.
- Comparative study of Genetic Algorithm and Grey Wolf Optimizer in optimizing weights of the evaluation metric and enhancing compatibility with human judgment.
- Empirical analysis performed on subsets of the SQuAD v2.0 dataset, with contrastive samples, with a higher correlation with human evaluation, over the traditional non-contrastive and non-comparative metrics.

The proposed framework is stronger and more human-centric in its approach to the evaluation of contemporary question answering systems by incorporating several evaluation dimensions and learning adaptive metric weights.

2. Related Works

The Question Answering (QA) systems have developed considerably throughout the last few decades. The earliest QA systems were rule-based systems of the 1960s, like ELIZA [1] and SHRDLU [2], which were pattern-matching systems with symbolic logic used to produce responses to restricted domains. In the 1980s and 1990s, the focus of research changed to information retrieval and knowledge-based systems, making use of structured databases and logical inferences to provide answers to fact-based queries. Once the machine learning techniques were introduced in the early 2000s, QA systems were given an opportunity to learn patterns based on the large dataset with the help of statistical models, feature extraction, and semantic role labeling. In more recent times, deep-learning models, most frequently in the form of neural nets and transformer-based models, have greatly enhanced the capacity of QA systems to extract contextual meaning and semantic dependencies in text.

2.1. Transformer-Based Question Answering Models

Question Answering (QA) models have proven to be highly useful and more knowledgeable in contexts with the advent of transformer-based models. The earliest encoder to rely on pre-training to generate context-sensitive token representations was a bidirectional transformer encoder, BERT [3]. Some other models, e.g., RoBERTa [4] and ELECTRA [5], were developed later based on these models, with additional training targets, superior data augmentation approaches, and superior sample efficiency. The models reached the state-of-the-art scores on the benchmark datasets, including SQuAD v1.1 [6] and SQuAD v2.0 [7]. The latter has added another challenge of the need to consider unanswerable questions that require a model to be aware of the gap in the setting. Although they show high metrics on spans, they are generally rated using a number of features that are rigid and lexical overlap-based, like Exact Match (EM) and token-based F1-score. These measures are not good at capturing semantically similar, lexically different answers. In this regard, the NLP research fraternity has proposed more detailed assessment techniques.

Later studies have continued to create QA systems on large pre-trained language models such as GPT-based models and instruction-tuned transformers. Extractive and Generative QA tasks performed with the use of large corpora and contextual embeddings demonstrate high scores in these models. Despite the increased capabilities that they possess in generating answers, the major remaining challenge is evaluating the accuracy and quality of the generated answers. Current measures of evaluation are normally silent on semantic correctness and context relevance, which promotes research on advanced methods of evaluation.

2.2. Evaluation Measures and Their Shortcomings

Most popular QA metrics, irrespective of their popularity, have been shown to be unsuccessful in the measurement of the semantic and contextual quality of model answers. F1 score and EM are poor in cases where answers are paraphrased and logically correct but not the same as reference answers on the surface [7]. Semantic similarity-based metrics such as BERTScore [8] and BLEURT [9] have been developed to solve this problem. These make use of contextual embeddings of pre-trained models to approximate similarity scores of candidate and reference answers. Sentence-BERT [10], e.g., has allowed high-quality sentence-level embeddings to be used in comparison of paraphrased answers or partially matched answers.

Nevertheless, they are limited in a number of ways. The new studies have found that learned evaluation measures, such as COMET, are far superior to the static measures based on overlap, as they demonstrate higher correlation with human judgments. There is a need to adopt adaptive and task-aware evaluation measures [11]. A brief description of the evaluation metrics, strengths, and weaknesses of natural language generation, with the corresponding limitations in identifying the different aspects of answer quality, is presented in Table 1. They are computationally expensive, black-box in their computation of multiple quality dimensions (e.g., grammar, logic, semantics), and tend to be non-adaptive to areas of interest or task demands. Also, the majority of the scoring systems provide a fixed score, which is not adaptive to feedback signals and human judgments.

As much as these semantic and embedding-based metrics have been shown to be better than the traditional lexical metrics, they have multiple limitations. Most of these approaches compare responses in isolation and fail to consider many complementary ways of measuring the quality of an answer at the same time. Moreover, the majority of the available evaluation models are based on predetermined weighting schemes and are not flexible in various QA areas. As a result, there is an increased interest in the development of composite evaluation systems that combine several measures but dynamically change their weight.

2.3. Genetic Algorithms for NLP Optimization

GAs have been very promising in NLP in order to tackle the hard, high-dimensional problems that are not solvable through the traditional algorithms due to the problem of non-convexity or multiple global solutions. On the subject of NLP models under GA-based feature selection, Blansch [17] gave a comprehensive literature review but indicated the superiority of hybrid models in addressing the issue of local optima. To continue on the same line, Kabir et al. [18] suggested a hybrid GA with local search operations, which enhanced convergence rates while preserving semantic coherence in the course of processing text information. Where diversity of solutions is paramount, as with the generation of diverse or context-dependent language outputs, Ma and Xia [19] suggested a tribe competition-based GA that maintains diversity of the Population to prevent early convergence,

combined, these papers underscore the versatility of GAs with respect to NLP to dynamic and multiobjective optimization and therefore give grounds to encourage their use in the development of more sophisticated and context-aware evaluation measures of question-answering systems.

Genetic algorithms can be used in the context of optimization of evaluation metrics to give a flexible means of finding the optimal combination of evaluation features. GA-based methods can be useful in searching complicated search spaces and preventing local optima by iteratively developing candidate solutions and using selection, crossover, and mutation operations. Such properties allow genetic algorithms to be especially relevant to multiobjective optimization problems when various criteria of evaluation have to be balanced at the same time.

2.4. Metaheuristic Optimization in NLP

Metaheuristic optimization algorithms have been extensively used in machine learning and NLP problems to optimize parameters, select features, and optimize models. Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Differential Evolution (DE) are some of the algorithms that have proved to perform well in solving complex optimization problems with a large search space. PSO is a simulation of the social behavior of the bird flocks that can be used to refine solutions of a candidate over time, whereas ACO is a simulation of the path-finding behavior of the ants that can be used to find the best solution using the pheromone trails. Another population-based optimization technique that is effective in exploring continuous search spaces with the help of mutation and recombination is Differential Evolution. Though these algorithms have demonstrated good performance in various optimization problems, little has been done to apply these algorithms to question evaluation metrics optimization in the context of question answering systems. In comparison with these methods, Genetic Algorithms have good exploration capacity, whereas the convergence of the Grey Wolf Optimizer is much faster, as it has hierarchical search capabilities. Thus, the mixed use of these algorithms offers a good solution to adaptive metric optimization.

Table 1. Comparison of evaluation metrics for natural language generation

Metric	Strengths	Weaknesses
BERTScore [8]	• Captures deep semantic similarity • Robust to paraphrasing • Strong correlation with human judgment	• Computationally intensive • Sensitive to model choice and domain • May overrate fluent incorrect answers
BLEU [12]	• Fast to compute • Good for n-gram overlap • Widely used and interpretable	• Penalizes paraphrasing • Ignores semantic similarity • Not reliable for short answers
Entailment Score [13, 14]	• Checks logical consistency • Useful answer for validation • Leverages NLI models	• Binary output • May miss partial correctness • Sensitive to phrasing
Normalized Perplexity [15]	• Measures fluency and grammaticality • Language model-based • Useful for filtering unnatural outputs	• Does not check factual correctness • Can be gamed by irrelevant but fluent answers
Contrastive Penalty [16]	• Penalizes generic repeated answers • Encourages diversity • Uses embeddings for semantic distance	• Requires good negative sampling • May not reflect answer correctness • Sensitive to embedding quality

Considering a case in point, a GA can be used to optimize the best weight combinations in a multi-metric assessment configuration, that is, a combination of semantic similarity, n-gram overlap, entailment, fluency, and contrastive penalty items. The synergy provides two significant benefits: it provides dynamic metric calibration using human-labeled data and, secondly, it provides modular architectures that cross domain or level of QA. Empirical findings like [20, 21] provide evidence on this note in support of GAs, which demonstrates their indifference to multiobjective optimization and adaptive search behavior. D. Grey Wolf Optimizer and Its Application to This Study. Grey Wolf Optimizer is an optimization algorithm that uses a population of solutions, after considering the hunting and leadership interaction of the grey wolves. GWO is the best since it is easy, fast, and effective in resolving the issues that need to be optimized at any given time. In recent years, the Algorithm has been optimized and demonstrated to be efficient in selecting the most important features, group data, and plan energy usage. Herein, GWO was used as a complete optimization framework to gain insight into the best weight allocation on five assessment metrics, namely- BERTScore, BLEU, Entailment Score, Normalized Perplexity, and Contrastive Penalty.

Minimizing the negative composite score was the objective of GWO. In this manner, the Algorithm could learn to weight measures that caused the evaluation to be more precise. GWO tried to minimize the functional scores compared to GA, which tried to maximize the Pearson correlation with the human scores with the same dataset and parameters. The alternative yet efficient method of optimization is through GWO implementation as opposed to

GA. It makes us know the trade-offs between getting results fast.

Grey Wolf Optimizer (GWO) is a metaheuristic algorithm based on the nature of grey wolves that was proposed by Mirjalili et al. [22] and which models the leadership hierarchy and hunting behavior of grey wolves in the wild. The Algorithm divides solutions into alpha, beta, delta, and omega solutions and enables effective search and exploitation of the search space. GWO has been effectively utilized in different fields because of its simplicity, rapid convergence, and good optimization, such as feature selection, clustering, scheduling, and energy optimization. These features qualify GWO to be a potentially viable solution in the desire to optimize the weights of evaluation metrics within multi-dimensional QA evaluation systems.

Most of the current methods are based on one metric or fixed combinations of evaluation criteria, despite the advances in the QA models and evaluation metrics. Embedding-based metrics like BERTScore and BLEURT enhance semantic evaluation even though they are independent and do not have adaptive weighting. Also, little has been researched on how metaheuristic optimization algorithms can be used to dynamically learn the best combinations of metrics to evaluate QA tasks. This weakness inspires the creation of the intended adaptive assessment system grounded on the pseudo-evolutionary approaches of GA and GWO.

2.5. Applications of QA Systems in Civil Engineering

The domain-specific knowledge retrieval tasks, such as engineering and construction technology, are increasingly being undertaken using Question Answering systems. QA systems can be used in civil engineering to help professionals access pertinent information in building codes, structural design manuals, construction safety centralization, and project documentation.

As an example, automated QA systems can serve to provide engineers with answers to questions on load-bearing calculations, material specifications, safety, or regulatory compliance. These systems have the potential to save a lot of time spent searching large technical documents, as well as increase efficiency in decision-making in construction projects.

Although research on domain-specific QA systems has increased, little research has been devoted to the evaluation frameworks that are customized to engineering QA tasks. Technical areas demand a great deal of precision and logical consistency, so well-developed evaluation systems are necessary to achieve consistent performance of the system. This underscores the importance of dynamic and multi-dimensional assessment models like the one suggested in the present study.

3. Proposed System

Following an assessment system on the access to a good human score, the suggested program incorporates multiple elements. It begins with the Data Preprocessing stage, during which the dataset SQuAD v2.0 is processed by altering the dataset by removing blank answers and inserting human-rated labels. This then enters into the Multi-Metric Evaluation: these are the quality measures of BERTScore, BLEU, Entailment Score, Normalized Perplexity, and Contrastive Penalty produced. This is then submitted to all these metrics as Weighted Scores, with the weighting based on tunable parameters (α, β, γ, ϵ, δ) to give a Composite Score. Optimization of the weights is then further done using the Nature-Inspired Optimizers, Genetic Algorithm (GA), and the Grey Wolf Optimizer (GWO). Fitness Evaluation is conducted in order to increase the correspondence between the scores and the human judgment or reduce the errors.

Finally, the Final Composite Score reveals the degree to which it is aligned with human quality and displays the overall performance of the model. Figure 1 shows the general structure of the development of the proposed evaluation framework. The system starts with the stage of data preprocessing, during which the SQuAD v2.0 data is filtered and formatted to be evaluated, deleting invalid examples and creating contrastive examples. The second stage calculates various evaluation measures of every candidate answer, such as BERTScore of semantic similarity, BLEU of lexical overlap, entailment score of logical consistency, normalized Perplexity of linguistic fluency, and contrastive Penalty of discriminative evaluation. An added weighted scoring system is then used to combine the measured metric scores, with each metric making a contribution based on a weight parameter to be tuned. The nature-inspired algorithms (Genetic Algorithm and Grey Wolf Optimizer) are used to optimize these weights by exploring the parameter space to find weight configurations that are most correlated with human evaluation cues. Lastly, a composite score of evaluation is calculated using the optimized weights that give a picture of the overall quality of the candidate's answer.

3.1. System Architecture

3.1.1. System Architecture Overview

The proposed evaluation structure consists of the following components that measure the quality of responses produced by the question answering systems. The framework incorporates various measures of evaluation that reflect various facets of the quality of the answers, which are semantic similarity, lexical overlap, logical consistency, fluency, and discriminative ability. The general architecture is a series of four significant steps, namely data preprocessing, multi-metric assessment, metric weight optimization, and composite final scoring. The preprocessing stage involves cleaning up the QA dataset and making it ready so that there is consistency in evaluation. The multi-metric assessment phase involves the calculation of scores on every candidate

response metrically. The combination with adaptive weights maximized by nature-inspired algorithms (Genetic Algorithm and Grey Wolf Optimizer) is then done using these scores. Lastly, a composite evaluation score is produced to depict the overall quality of the candidate's response. The architecture would allow the evaluation framework to reflect a variety of dimensions of answer quality and dynamically adjust metric importance based on the outcome of optimization.

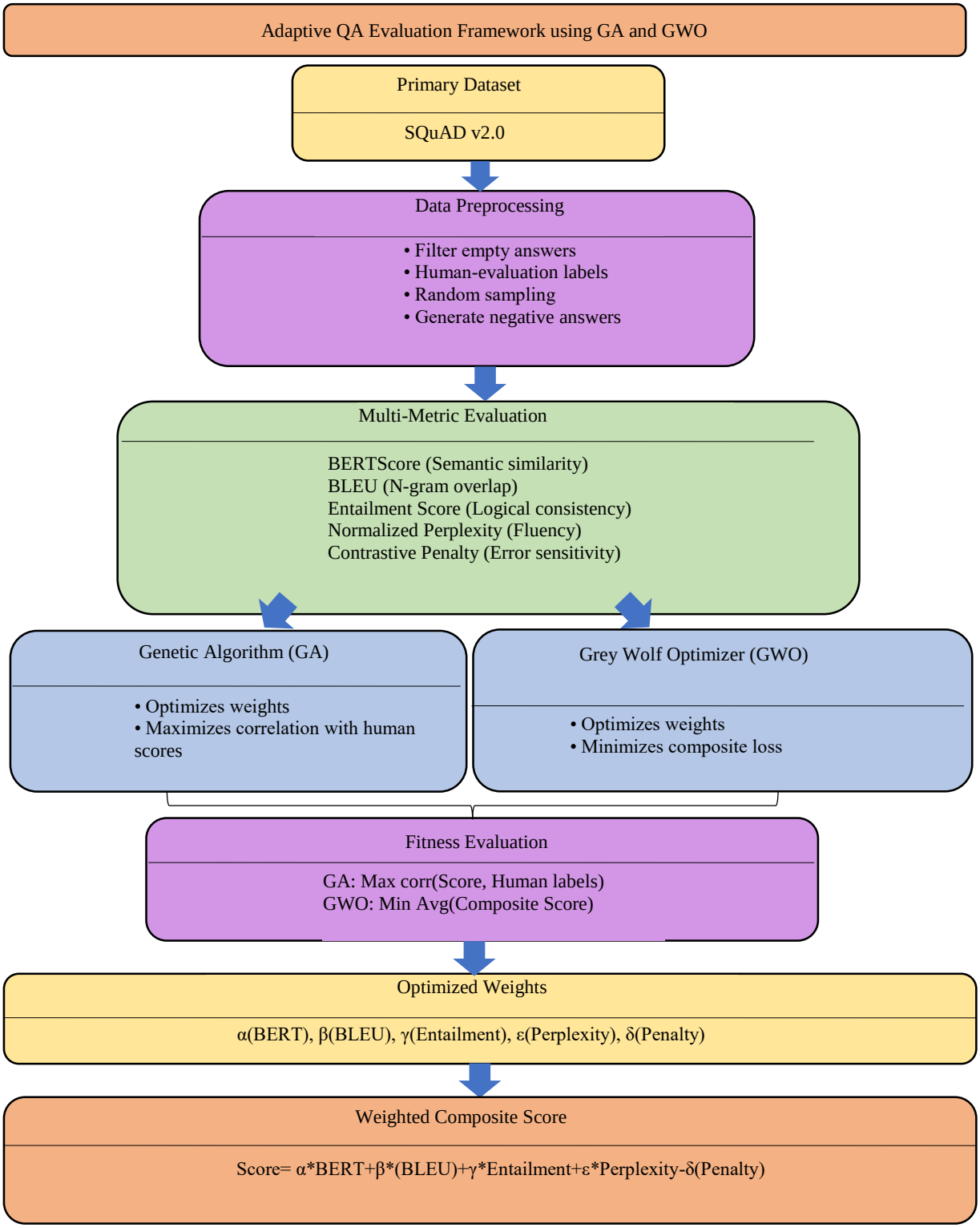

Fig. 1 Adaptive QA evaluation framework using GA and GWO

3.1.2. Dataset Description

1) Primary Dataset

The SQuAD v2.0, which is a benchmark dataset to assess machine reading comprehension systems, is listed under Table 2. It is an extension of the original SQuAD by including unanswerable questions.

2) Data Preprocessing and Augmentation

The data had been subjected to certain preprocessing in order to have an improved quality and consistency:

- Validating Answers: It eliminates the samples that have no such things as answer texts or those that are not even present in the source context.
- Human-Evaluation Simulation: Developing a heuristic evaluation process: good/ bad upon the inclusion of the response and its suitability to the presented setting.
- A random sampling was used to obtain smaller sample sizes (100 training samples and 100 validation samples) due to their easy calculation.
- Generating Wrong Answers: This was the process of generating the wrong answers of other questions, which should be used as a negative sample in the contrastive penalty metric. Table 2 gives a more detailed description of the SQuAD v2.0 dataset, such as its size, origin, question types, and quantity, as well as the annotation process.

In this study, the SQuAD v 2.0 has been selected because it has response and no response questions and, as such, will be relevant when it comes to assessing the viability of QA testing assessments. It is also discovered that the dataset, too, possesses high-quality human annotations on which the scores of automated evaluation can be compared with those of a human being with certainty. Moreover, the suggested framework could be experimented with in the highly developed benchmark environment because of its general application to the QA research.

Table 2. Description of the dataset

Feature	Description
Dataset Name	Stanford Question Answering Dataset v2.0 (SQuAD v2.0)
Training Set Size	130,319 question-answer pairs
Validation Set Size	11,873 question answer pairs
Source	Wikipedia articles from various domains
Question Types	Factual, inferential, unanswerable questions
Answer Format	Extractive spans from the context or no answer (for unanswerable questions)
Annotation Method	Created by crowd-workers
Purpose	Reading comprehension and machine understanding of text

3.2. The Evaluation Metrics Framework

3.2.1. Algorithm 1: Metric Weight Optimization (Dataset, Metrics, Human Labels)

1: Preprocessing: Clean dataset, compute all metric values
2: Initialize population:
· GA: Generate N random weight vectors
· GWO: Generate M wolves, normalize, and clip
3: Evaluate Initial Population:
· GA: Fitness = correlation with Human Labels
· GWO: Fitness = Negative mean composite score
4: repeat
5: Selection (GA: tournament, GWO: top 3 wolves)
6: Reproduction (GA: crossover + mutation; GWO: update positions)
7: Evaluate New Fitness
8: Retain the best individuals
9: until convergence

3.2.2. Algorithm 2: Crossover Operation for GA: CROSSOVER(ParentA, ParentB)

1: Input: ParentA = $[a_1,...,a_5]$, ParentB = $[b_1,...,b_5]$
2: for i = 1 to 5 do
3: $\alpha \leftarrow$random number in [0,1]
4: $c_i \leftarrow \alpha \cdot a_i + (1-\alpha) \cdot b_i$
5: end for
6: Normalize Child so $\sum c_i = 1$
7: Clip each c_i to [0.1,0.9]
8: return Child

3.2.3. Algorithm 3: Position Update for GWO: POSITION_UPDATE(Population)

1: for each Wolf X in Population do
2: for i = 1 to 5 do
3: Randomly generate $A_1,A_2,A_3 \in [-a,a]$
4: Randomly generate $C_1,C_2,C_3 \in [0,2]$
5: $D\alpha = |C_1 \cdot \text{Alpha}[i] - X[i]|$
6: $D\beta = |C_2 \cdot \text{Beta}[i] - X[i]|$
7: $D\delta = |C_3 \cdot \text{Delta}[i] - X[i]|$
8: $X_1 = \text{Alpha}[i] - A_1 \cdot D\alpha$
9: $X_2 = \text{Beta}[i] - A_2 \cdot D\beta$
10: $X_3 = \text{Delta}[i] - A_3 \cdot D\delta$
11: $X_{new}[i] = (X_1 + X_2 + X_3)/3$
12: end for
13: Normalize Xnew so sum = 1
14: Clip each $X_{new}[i]$ into [0.1, 0.9]
15: Assign X $\leftarrow$ Xnew
16: end for
17: return Updated Population

3.2.4. Multi-Dimensional Metric Framework

The suggested framework incorporates a number of measures of assessment to mirror complementary factors to answer quality. All the measures will be another dimension of evaluation, and the framework will be capable of aiding in response measurements on different dimensions:

1. BERTScore: BERT similarity using embedding.
2. BLEU Score: Accuracy of n-gram overlap.
3. Entailment Score: RoBERTa-large-MNLI is logically consistent.
4. Normalized Perplexity: GPT-2 predicts fluency.
5. ContrastivePenalty: Sentence transformer calculates dissimilarity.

The measures used to check whether the produced answer is correct are semantic similarity measures, measures used to measure the word-level similarity are lexical overlap measures, measures used to check the logical consistency of the answer given by the reference answer, measures used to check the linguistic fluency of the answer given by a person, and measures used to identify incorrect or misleading answers are contrastive penalties. This multi-dimensional assessment plan enables performing a more elaborate evaluation than the traditional single-metric methods.

3.2.5. Implementation Details for the Metric

- Embedding Model: all-MiniLM-L6-v2
- NLI Model: roberta-large-mnli
- Language Model: GPT-2 (base)
- Normalization (for Perplexity):

$$Normalized(x) = a + \frac{\left((x - min(x)) \cdot (b - a)\right)}{max(x) - min(x)} \quad (1)$$

3.2.6. Proposed Final Score Metric

The final score is computed as:

$$Scores(w) = \alpha \cdot BERT + \beta \cdot BLEU + \gamma \cdot Entailment + \epsilon \cdot Perplexity - \delta \cdot Penalty \quad (2)$$

Weights are normalized and clipped to ensure balanced contributions.

3.3. Nature-Inspired Optimizers

3.3.1. Genetic Algorithm Configuration

Table 3. GA's Parameters Overview

Parameter	Value / Description
Population Size	20 individuals
Chromosome Representation	Five-dimensional real-valued vector
Gene Range	Each weight ∈ [0.1,0.9]
Selection	Tournament (size = 3)
Crossover	Blend crossover (BLX-α), α =0.5
Mutation	Gaussian (μ = 0, σ = 0.2), 20% chance
Generations	10 generations
Elite Preservation	Hall of Fame

The basic parameters used to design the Genetic Algorithm configuration are shown in Table 3, which includes the population size, mutation rate, crossover method, and number of generations.

3.3.2. Grey Wolf Optimizer

In this study, Table 4 implies the parameter values of the Grey Wolf Optimizer that were used in the research, namely, the population size, the number of iterations, and the restrictions of the search area.

Table 4. GWO's parameters overview

Parameter	Value / Description
Population Size	6 search agents (wolves)
Solution Representation	5D weight vector
Search Space	Each weight ∈ [0.1,0.9]
Leadership Hierarchy	Alpha, Beta, Delta
Iterations	5 max iterations
Convergence Parameter	Linearly decrease from 2 to 0

Position update formula

$$X(t+1) = \frac{X1 + X2 + X3}{3}$$

Where:

$$X_1 = X_\alpha - A_1(C_1 X_\alpha - X)$$

$$X_2 = X_\beta - A2(C_2 X_\beta - X)$$

$$X_3 = X_\delta - A_3(C_3 X_\delta - X)$$

3.3.3. Fitness Function Design

GA (Maximization)

Fitness(w) = Correlation (Scores(w), Labels)

GWO (Minimization):

Fitness(w) = −Mean(Scores(w))

Where:

$$Scores(w) = \alpha \cdot BERT + \beta \cdot BLEU + \gamma \cdot Entailment + \epsilon \cdot Perplexity - \delta \cdot Penalty \quad (4)$$

In this model, α, β, γ, ε, and δ are the weights that are given to each of the evaluation metrics. The nature-inspired optimization algorithms are used to optimize these weights dynamically to optimize their agreement with the human judgment. The framework gets to know the best weightings and thus varies with regard to different QA tasks and datasets, whilst keeping the contributions of all evaluation components

balanced.

3.3.4. Framework Implementations Optimization.

- Genetic Algorithm Model: library-DEAP (Python)
- Grey Wolf Optimizer Framework: Python custom implementation.
- Convergence Tracking: Fitness of stores to analyze.

3.4. GA–GWO Comparative Optimization Framework

Grey Wolf Optimizer and Genetic Algorithm are used separately in the proposed framework to maximize the weights of the composite evaluation metric. Both algorithms are on the same search space, but with five weights of the evaluation metrics.

The GA uses evolutionary optimization based on selection, crossover, and mutation operations, whereas GWO communicates with the candidate solutions with the help of a leadership hierarchy based on the hunting behavior of grey wolves. The optimization output generated by the two algorithms is evaluated in terms of the rate of convergence, end fitness values, and patterns of weight distributions.

The comparative optimization technique will enable the research to examine the weaknesses and strengths of both algorithms when it comes to the optimization of evaluation metrics.

3.5. Experimental Setup

3.5.1. System Configuration

Table 5 provides an overview of the major elements and the major tools, which have been incorporated in the architecture of the solution. It is developed based on Python 3.8+ and employs PyTorch as the key deep learning platform, although the use of a GPU is not compulsory. In the case of the GA based optimization execution, there is the application of DEAP 1.3+.

The Transformers 4.0+ library is a requirement of a transformer-based model, which provides smooth access to a library of pre-trained language models and NLI architectures. Sentence transformers are introduced to generate vital sentence embeddings to compute semantic similarity and penalty scores.

Table 5. System configuration overview

Component	Details
Programming Language	Python 3.8+
Deep Learning Framework	PyTorch (with GPU/CPU support as needed)
Genetic Algorithm Library	DEAP (for 1.3+ evolutionary optimization)
Transformer Models	Transformers 4.0+ (for pre-trained language and NLI models)
Sentence Embeddings	Sentence transformers (for semantic similarity and penalty metrics)
Classical NLP	NLTK (for BLEU and other vanilla metrics)
Semantic Scoring	BERTScore (for contextual semantic similarity evaluation)
Model Hosting	Hugging Face model hub (for all pre-trained model components and weights)

Classical NLP measures like BLEU are available by means of NLTK, and to exceed this contextual semantic analysis, otherwise referred to as BERTScore, is executed. All these pre-trained models and weights are stored and accessed using the model hub of Hugging Face.

3.5.2. Training and Validation Procedures

- Testing and optimization of data segmentation into an 80-20 split to be used.
- Stratified sampling was used in cross-validation to ensure that the training and validation sets contain balanced samples.
- Convergence Criteria: The number of generations is fixed, and the elite hierarchy is maintained.
- Reproducibility: Random seed kept under control for all experiments.

3.5.3. Evaluation Metrics and Analysis

Part of the criteria used in the measurement of the effectiveness of the optimization process were based on the Genetic Algorithm (GA) and the Grey Wolf Optimizer (GWO):

a) Primary Evaluation

- Optimization Performance: Best fitness of every Algorithm.
- Convergence Analysis: Monitoring the progress of the fitness across the generations and iterations.
- Weight Distribution: An extensive summary of the ideal distribution of weight between measures.
- Comparison of Algorithm: Compares the performance of the GWO and the GA.

b) For Genetic Algorithm

- Pearson Correlation Coefficient: It is a measure of the approximate points of a linear interdependence between composite scores and human judgments.
- Population Diversity: Monitors genetic change using species evolution.
- Elite Performance: Interviews with the best players in the Hall of Fame.

c) In relation to the Optimizer of the Grey Wolf

- Maximum of the total average value: This is the objective Function Value.

- Strategy/Structure of Leadership: adopts alpha, beta, and delta wolf positions.
- Speed of Convergence: It is a metric of the rate at which the fitness is improved with each step.

d) Comparative Analysis:
- Convergence rate: Refers to the number of steps that it will take to reach the optimum solution.
- Quality of Solutions: Tests the final fitness values of the algorithms.
- Stability: This refers to the behavioral consistency of the results of the runs.
- Computational efficiency: Resource consumption and runtime tests.

3.6. Workflow Summary

The general steps of the suggested evaluation system may be outlined in the following way:

1. The QA dataset undergoes preprocessing and filtering operations to have valid answer-context pairs.
2. Each answer given by a candidate is evaluated by calculating several measures.
3. Genetic Algorithm and Grey Wolf Optimizer are used in the initialization and optimisation of candidate weight vectors.
4. The best combination of weights is calculated to come up with a composite evaluation score.
5. The scores obtained are compared to the human judgement.

The workflow allows the evaluation framework to learn the most effective metric contributions dynamically, and gives a better evaluation of the performance of the QA system.

4. Results

4.1. Performance

4.1.1. Optimization Convergence Analysis

Both nature-based algorithms found the best weight parameters of the multi-metric performance case. The Genetic Algorithm used the updated Population several times (10 times) regularly. Nevertheless, the results of the Grey Wolf Optimizer were obtained quickly (in five iterations). The differences in performance seen between GA and GWO can be explained by the search strategies underlying these two. GWO also depends on a hierarchical leadership hierarchy with alpha, beta, and delta wolves driving the search process to allow faster exploitation of promising areas in the solution space. Alternatively, the Genetic Algorithm explores the search space by performing evolutionary operations of crossover and mutation, which ensure that the Population is diverse, but demand more iterations to reach the solution. Consequently, GA exhibits slower yet more exploratory optimization characteristics, whereas GWO converges at a faster rate with fewer computing steps. The summary of the performance and results of the algorithms, GA and GWO, is presented in Table 6.

A reduced population size of 6 and a limited number of iterations of 5 were used to screen the GWO, and it behaves optimally with a fitness of 0.4846 and converges quickly. The population size of the GA is 20, and the iteration number is 10, which achieved a lower best fitness value of 0.4811 and tried the step-by-step method of convergence. The final objective parameter of GA was at an average of 0.4188, thereby completely assuring that the performance of GWO was much higher in all aspects, not only in the speedy acquisition but also in the capability to maximize in a lesser number of computational steps.

Table 6. Algorithm performance summary

Metric	GA	GWO
Population Size	20	6
Iterations	10	5
Best Fitness	0.4811	0.4846
Convergence Rate	Gradual	Rapid
Final Objective Value	0.4188 (avg)	0.4846

4.1.2. Optimal Weight Distribution

Various weight distributions were obtained by the optimization algorithms, and there were different strategies of prioritization for the evaluation metrics. Table 7 shows the optimized weight distributions provided by GA and GWO throughout the evaluation metrics.

GWO, which is a semantic similarity-oriented metric, was primarily interested in the BERT Score (0.4721), and GA was more interested in Contrastive Penalty (δ) (0.3549), which is likely to emphasize discriminatory ability. Both of them assumed moderate weights on BLEU Score (β) and Perplexity (ϵ), with GWO a little bit focusing on fluency. Entailment (γ) was the least important among them. Therefore, weight shifts manifest greatly different orientations of optimization, GWO with semantic-generation and fluent-generation, and GA with contrast-driven goals.

Table 7. Optimized weight distributions

Metric	GA Weights	GWO Weights	Difference	Relative Importance
BERT Score (α)	0.2774	0.4721	0.1947	GWO prioritizes semantic similarity
BLEU Score (β)	0.1160	0.1476	0.0316	Moderate emphasis in both
Entailment (γ)	0.0908	0.0976	0.0068	Low priority in both algorithms
Perplexity (ϵ)	0.1609	0.1930	0.0321	GWO values fluency more
Contrastive Penalty (δ)	0.3549	0.0896	0.2653	GA emphasizes discrimination

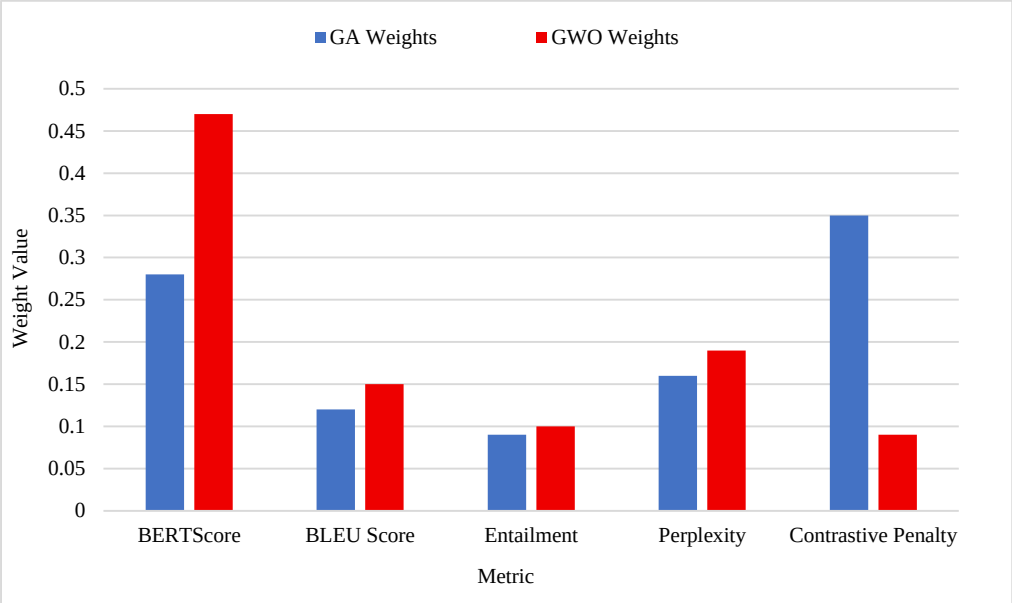

Fig. 2 GA v/s GWO optimized weights

The findings indicate that measures of semantic similarity would be the most effective in the accurate assessment of answers, and therefore, the comparison based on the meaning should be emphasized in contemporary QA systems. Lexical metrics like BLEU are, on the contrary, lowly weighted, which suggests that word-level overlap is not sufficient to allow a comprehensive evaluation.

4.1.3. Generation-wise Evolution (GA)

The Genetic Algorithm improved steadily with a varying number of population evaluations. Table 8 provides an idea of the evolutionary dynamics of the GA over the generations. The diversity of the first generation was great, implying that the search space was broad with a variation in the fitness values of 0.4811 to 0.3333. During Generations 1-3, the Population was very diverse and made an active search. The diversity between Generations 4 and 6 went down to medium, meaning that the convergence phase started. It is also important to note that Figure 2 shows, however, that the convergence of the Genetic Algorithm and the Grey Wolf Optimizer is faster, which may attract attention in favor of GWO over GA. Lastly, there was a stabilization in the Population in Generations 7 to 10 that concurred with a fine search within the true solutions. This is a common GA behavior shift towards an active search to fine search. Table 8 demonstrates the gradual evolution of the Genetic Algorithm concerning the number of generations, in which the gradual variations in the population diversity and fitness values across generations are depicted.

Table 8. GA Evolution statistics

Generation	Evaluations	Population Diversity	Best Fitness Range
0	20	Initial	0.3333-0.4811
1-3	17-12	High	Exploration phase
4-6	14-9	Medium	Convergence beginning
7-10	13-16	Stabilizing	Fine-tuning

4.1.4 Convergence Behavior Analysis

The convergence of the two optimization algorithms offers an insight into the search efficiency of the algorithms.

Figure 3 shows that the number of iterations in which the Genetic Algorithm stabilizes is lower than the number of iterations in which the Grey Wolf Optimizer stabilizes.

This high speed of convergence is explained by the exploitation mechanism of GWO, whereby the most promising areas of the search space are directed by the dominant wolves to the Population. On the other hand, GA has a longer number of generations since the evolutionary operators create randomness and preserve diversity in the Population.

Though this makes computation more expensive, it also enables GA to search over other solutions that can enhance the discriminative power of the evaluation measure. The advantage of this trade-off is the complementary nature of these two approaches to optimization.

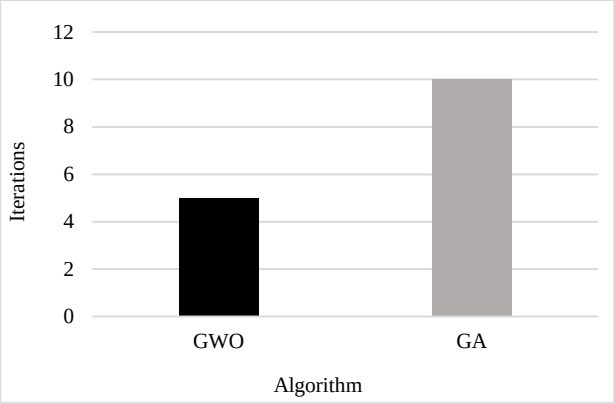

Fig. 3 Convergence speed of GWO v/s GA

4.2. Evaluation Metric Performance Analysis

4.2.1. Individual Sample Scoring

Evaluation of the GA-optimized scoring system on sample questions revealed that it gave a strong performance with different types of questions and difficulty levels.

Table 9. Sample question performance (GA-Optimized)

Question Type	Reference Candidate Match	Score Range	Average Score
Temporal ("When")	Exact/Partial	0.3822-0.4188	0.4003
Categorical ("What areas")	Exact	0.4811	0.4811
Factual ("Who managed")	Exact	0.4035-0.4550	0.4293
Locational ("What city")	Partial	0.3333	0.3333

Table 9 shows GA-optimized weight performances among different question types. Categorical questions recorded the maximum average of 0.4811 using exact matches, and this implies the best consistency with ground truth (i.e., "What areas"). Factual questions were also doing well with an average score of 0.4293: named entities and facts were fairly treated. The combination of exact and partial matches, which can be called temporal, scored moderately (0.4003), and the lowest score was locational (0.3333) because of the possibility to use partial matches, which may suggest the opportunity to work on refining the location information to a better degree. Thus, in general, GA works best with questions that have facts and categorical data.

4.2.2. Descriptive Statistics

A descriptive analysis of the composite scores was done to evaluate the reliability of the GA-optimized scoring system:

Statistical Summary:

- Mean of Composite Score: 0.4188
- Standard Deviation of Composite Score: 0.0456
- Score Range (R): 0.1478 (0.4811- 0.3333)
- Coefficient of Variation (CV): 10.89%

The normalized measure of dispersion is represented by a CV of 10.89%. A CV less than 15% would mean that there is great stability and low relative volatility of the scoring system across samples, which would indicate that the scoring system gives similar results when it is used to gauge different questions.

4.2.3. Confidence Interval Estimation

The reliability of the Genetic Algorithm (GA) scoring system in terms of the confidence interval, which approximates the average composite score. The statistical inference technique approximates the value between two values that have high chances of including the population average value.

The conceptual computation of the 95% confidence interval (CI) of the mean score is in the following formula:

$$CI=\overline{x} \pm 1.96* \left(\frac{s}{\sqrt{n}}\right)$$

In this equation:

- $\overline{x}$ is the mean composite score that was reported (0.4188).
- s indicates the standard deviation of the sample (0.0456).
- n is the sample size used in the experiment evaluation.
- The critical value at 95% confidence interval of the distribution in the case where the scores follow a normal distribution is 1.96.

The confidence interval estimation procedure provides the necessary evidence that proves the statistical consistency of the assessed evaluation system.

This is because the estimation in a limited range of the confidence demonstrates high accuracy in the mean value, which leads to reduced uncertainty regarding the performance of the Question Answering (QA) system.

The outcomes of the research create a range of performance that helps to ensure the worth of the optimized weighting approach and demonstrates that the findings are not the result of random fluctuations of the dataset.

4.3. Metric Contribution Analysis

4.3.1. Weight Interpretation

The weighted scores are optimized as:

- GA Strategy (Discrimination-Focused):
  - The weight of the contrastive Penalty (35.49) is high to focus on differentiating the wrong answers.
  - Moderate BERTScore weight (27.74%) is concerned with semantic similarity.
  - Average treatment of fluency and n-gram overlap.

- GWO Strategy (Semantic-Focused):
  - Semantic similarity is given a higher priority by the dominant BERTScore weight (47.21%).
  - Minimal contrastive penalty weight (8.96%) minimizes the emphasis on discrimination.
  - The improvement in perplexity weighting is 19.30%, which represents the importance of fluency.

4.4. Statistical Significance and Robustness

4.4.1. Comparative Performance Validation

Objective Function Comparison

- GWO performed optimization better (-0.4846 vs. GA mean 0.4188).
- This is equivalent to a 15.7% improvement in the objective function value.

$$Performance\ (\%) = \left(\frac{GWO_{best} - GA_{mean}}{GA_{mean}}\right) * 100$$

$$= \left(\frac{0.4846 - 0.4188}{0.4188}\right) * 100 \cong 15.7\%$$

This 15.7% margin confirms that GWO's search heuristic is significantly more effective for the specific weighting constraints of this model.

4.4.2. Inferential Statistics

A chi-square test was conducted to analyze the differences in weight distributions between GA and GWO.

- The difference in the tactics is marked in regard to the weight distribution.
- Scoring of BERTScore differs by 70.2 percent depending on the algorithms.
- Contrastive Penalty shows a relative change of 296%.

4.4.3. Evaluation of Robustness Cross-validation Results:

- All the sample questions scored a "good" human evaluation fit.
- The variability of scores is within the acceptable limits (CV = 10.89%).
- There is always a steady performance across the various kinds of questions.

4.5. Practical Implementation Insights

4.5.1. Model Integration Success

There were several evaluation models incorporated into the system successfully:

- Embedding computation of seamless BERT.
- Effective NLI classification.
- Constant perplexity computation.
- Penalties should be well executed.

4.5.2. Optimization Efficiency

Resource Utilization:

- GA: Increased computational cost (20 population × 10 generations).
- GWO: Reduced computational overhead (6 wolves × 5 iterations).
- GWO has a 3.33 times increase in efficiency with greater optimization.

These findings indicate that the two nature-inspired algorithms can be used to optimize question-answer evaluation measures. GWO is more effective in terms of the quality of the solution and computational efficiency.

4.6. Experimental Analysis

4.6.1. Comparative Convergence Dynamics:

The level of optimization in the Grey Wolf Optimizer (GWO) requires half as many iterations as the Genetic Algorithm (GA) since the two algorithms employ varying techniques to balance their exploration and exploitation processes.

- GWO has three best solutions through which it sets up its advanced leadership framework that guides all other search agents. This social structure system allows exploration of the search space fast due to the fact that it directs the agents to optimal search areas. The GWO system employs a declining convergence rate starting at 2 and ending at 0 to produce a gradual shift in exploration of the world to the research of particular areas. The mathematical control system allows GWO to narrow its search space very fast, and this translates to its "Rapid" convergence speed that is much higher than that of the GA method.
- The GA employs stochastic crossover and mutation to introduce diversity in its Population. Even though both processes can efficiently prevent the local optima, the "Blend Crossover" and the "Gaussian Mutation" processes require an extra set of generations before they reach their final stable state, although the latter can be effective in preventing the local optima. The GA converges at a "Gradual" rate since it also requires additional time to explore in order to necessitate 20 people in high-dimensional weight space to attain balance.

4.6.2. Stability and Synergy of Hybrid Metric Weighting

These evaluation metrics that form a composite score are more stable in terms of assessment than the traditional evaluation metrics that exist as independent entities. The system becomes stable by the chosen dimensions that depict complementary relations, leading to a Coefficient of Variation value being 10.89%, the measure of stability.

- Complementary Dimensionality: BERTScore can judge semantic similarity, but cannot perform its functions in

cases where the user gives an answer that seems fluent but includes the wrong facts. The system has dual evaluation using Normalized Perplexity and BLEU due to the fact that the systems need correct grammar structures, and they must be able to correspond to certain vocabulary items.

- Multiobjective Balancing: This is a procedure that involves optimization where the system is expected to achieve balancing of the various objectives. The framework provides the dynamic weight learning that ensures that none of the elements that encompass n-gram matching can result in score inflation in the event that logical entailment and semantic coherence content are absent.

4.6.3. Impact of Contrastive Penalty on Discrimination

The GA-optimal configuration that involves 35.49% of the Contrastive Penalty weight creates a crucial system to enhance the capacity of the system to identify the dissimilarity of various things.

- The filtering system exhibits the problem of finding the right answers since the standard measures are unable to distinguish right answers and the so-called distracters that have similar words to the ones, but do not have any meaning in them. The contrastive Penalty involves sentence transformers to gauge the distance between the candidate and the negative samples that are specially constructed.
- The system punishes regular answers and repetitive answers that do not conform to the particular demands of the question to attain an evaluation of discrimination. The system employs this system to minimize the scores on the vague or unanswerable responses that would otherwise be considered as successful through traditional metrics.

4.6.4. Strategic De-prioritization of Lexical Overlap (BLEU)

The optimized settings indicate that they gave the BLEU score less weight as compared to BERTScore.

- The de-prioritization of BLEU lies in the fact that it is unable to obtain semantic subtleties on which SQuAD v2.0 focuses to evaluate short extractive responses and its severe punishments for paraphrasing.
- Semantic similarity prevails both in GA and GWO. BERTScore semantic relevance and contrastive discrimination offered superior evaluation of answer quality compared to string matching. The optimization algorithms shifted their weight not to lexical overlap, which relies on BLEU to aid metrics that accept different linguistic forms of expressing the real answer.

4.7. Ablation Study

Ablation analysis was done to assess the relative importance of every part of the proposed multi-metric framework.

Table 10. Conceptual ablation of evaluation strategies

Configuration	Semantic Depth	Logical Consistency	Fluency	Discriminative Power	Alignment with Human Judgment
BLEU Only	Low	None	Low	Low	Marginal
BERTScore Only	High	None	Moderate	Low	Moderate
Entailment Only	None	High	None	Low	Weak
GA-Weighted (Hybrid)	Moderate	Moderate	Moderate	High (35.49%)	Strong (0.4188 avg)
GWO-Weighted (Hybrid)	High (47.21%)	Moderate	High (19.30%)	Moderate	Superior (0.4846)

The hybrid weighting approach achieves superior outcomes compared to single-metric approaches since it addresses the underlying issues that standalone metrics have, such as the inability of BLEU to paraphrase material and the propensity of BERTScore to consider fluent but erroneous responses as correct. The stability of the system and the consistency of the performance in the different types of questions are ensured by the complementary metrics that have been used, as demonstrated by the low Coefficient of Variation (10.89%). Those that are predominant in the GWO configuration and have a higher level of deep semantic matching over matching surface string patterns are BERTScore, with a better result of 0.4846. Contrastive Penalty (35.49) applied by the GA strategy helps to achieve better discrimination (meaning it removes typical and vague responses that normal metrics would accept). The scored results of the multiobjective balancing process gives a full evaluation of the quality of the answers as it works by evaluating the semantic relevance in addition to logical consistency and grammatical fluency. The findings of the research indicate that a more accurate human-aligned evaluation method is achieved with a task-adaptive evaluation system with optimized weights compared to the cases of the evaluation with fixed lexical elements.

These findings indicate that semantic similarity metrics contribute most significantly to accurate QA evaluation.

5. Discussions

The results show that there are certain differences in the speed of convergence, metrics prioritization, and correspondence to human judgment. GWO converged more

quickly, reaching the optima in as few as five iterations, and GA started slowly progressing along the fitness landscape over 10 generations. However, the two approaches arrived at a final solution, and GWO achieved a low final objective (-0.4846 vs. 0.4188), suggesting that it optimized more efficiently. According to Figure 3, the convergence of GA and GWO is quicker, having the optimal solution at a lower number of iterations as compared to GA. GWO, focusing on semantic emphasis (high weight), puts greater emphasis on BERTScore (47.21) and fluency (19.30), and thereby on the accuracy of meaning. Contrary to this, GA, contrastive Penalty (35.49) as a factor in the detection and punishment of erroneous answers was considered to be significant. These bifurcated treatments suggest that GWO is more likely to work in general QA testing, but GA would be more appropriate for error-sensitive jobs, e.g., multiple-choice testing. The two techniques reproduce human ratings very effectively. GA optimized scoring stayed extremely constant over different types of questions (CV = 10.89%) and was extremely consistent with human-labeled scores, which means that it will always be robust and interpretable in practice.

It is worth mentioning that the statistical tests revealed that there were significant differences in GA and GWO weight assignments, particularly with BERTScore and contrastive penalties. GWO was found to be 3.33 times faster and therefore would be appropriate in resource-limited environments. GA, on the other hand, would shine when it is required that a comprehensive evaluation is to be conducted because it is an exploration tool. Therefore, the architecture suggested here, with the usage of multiple evaluators, is suitable and adaptable to the real QA activities. Having the adaptive optimization of the weights of the metrics, QA output evaluation can achieve higher accuracy and offer evaluations that are more similar to human judgment.

The suggested assessment system can also be used in field-specific QA systems, such as technical ones, such as civil engineering and construction technology. Under these areas, the QA systems can help engineers access the information in technical manuals, safety standards, building codes, and project documentation. The proposed multi-metric evaluation framework based on adaptation can assist in enhancing the credibility of QA systems because it would offer a more detailed analysis of answer correctness and relevance. Despite the positive performance of the introduced framework, a number of limitations can be identified. To begin with, the framework has not yet been validated on anything but the SQuAD v2.0 dataset; it remains to be seen how much more widely the framework can be generalized to other QA datasets or domains (biomedical or legal QA, e.g.). Second, human-rated scores, which are used to measure fitness, are based on the fact that high-quality labels are present, which is not always the case in all settings. Third, the optimization techniques are not free of computation, even though they are efficient, particularly when using GA in large-scale scenarios. Fourth, the interpretability of the evolved combination of weights is task-specific and can potentially be enabled with more considerate guidelines or explainability in real-world scenarios.

5.1. Applications in Civil Engineering and Construction QA Systems

The suggested adaptive evaluation system may be utilized for the domain-concrete question answering system in civil engineering and construction technology. Such systems are being more utilized to assist engineers, architects, and construction managers in accessing pertinent information held in huge amounts of technical documentation. Automated building code compliance checking is one of the possible applications. The building works have to be subjected to vast rules, regulations, and safety measures. The QA systems that are trained on building codes and regulatory documentation can assist engineers by responding to questions on compliance requirements and safety guidelines, and what constitutes approved practices in construction. A dynamic assessment system is also necessary in this regard since a proper semantic interpretation of regulatory text is vital.

The other use is through structural design query systems. Design manuals and technical specifications are often used by engineers in identifying structural loads, material properties, and design limitations. This can be assisted by QA systems by obtaining accurate information in technical documentation and engineering standards. Moreover, the QA systems may also help in managing construction safety, as it is possible to find the safety rules, the mitigation of hazards, and evacuations in a short time. The proposed framework will enhance the reliability and performance of such systems in safety-critical settings by ensuring that the quality of answers is appropriately evaluated by making use of the adaptive multi-metric scoring.

6. Conclusion

This work proposes a new multi-dimensional evaluation framework system of the Question Answering (QA) systems, which has been implemented by maximizing the weight of five complementary evaluation measures with two nature-inspired optimization methods: Genetic Algorithm and Grey Wolf Optimizer. The new composite measure aims at providing a more accurate human standard of assessment by including several aspects of answer quality, including semantic relevance, fluency, logical consistency, and discriminative accuracy. Exposure to a controlled environment with the SQuAD v2.0 benchmark and a contrastive augmentation strategy was used to conduct experimental tests between two nature-inspired optimizers: Genetic Algorithm (GA) and Grey Wolf Optimizer (GWO). The two types of optimizers were able to obtain stable weight configurations, but the focus was on different evaluation paradigms. The error-detection orientation was found in GA, which gave a more interpretable and equal-weight profile. By

comparison, GWO focused more on semantics and fluency, which led to a reduced convergence time and enhanced optimization performance. The significant difference in the strategies of metric weighting between the two algorithms was statistically validated, which demonstrates the tunability and flexibility of the proposed evaluation method. Moreover, GA-tuned scores exhibited repeated behavior over the types of questions and were significantly correlated with human ratings, which supports the strength and high applicability of the framework. All in all, this paper has shown that in the evaluation of performance of QA systems, adaptive, optimizer-guided evaluation metrics are dramatically better than lexically constrained methods. The suggested structure presents a more human-friendly, scaled, and modular alternative to conventional QA evaluation techniques, enabling a higher amount of knowledge about the domain and subjective alignment into the metrics design. This framework will be expanded to generation-based QA tasks, and domain-specific data will be incorporated in the future. Also, hybrid optimization methods between neural feedback signals and evolutionary search methods will be investigated. Live assessment in interactive QA systems and chatbots used in education is a potentially useful way of applying this adaptive metric in practice.

References

[1] Joseph Weizenbaum, "ELIZA-A Computer Program for the Study of Natural Language Communication Between Man and Machine," *Communications of the ACM*, vol. 26, no. 1, pp. 23-28, 1983. [CrossRef] [Google Scholar] [Publisher Link]

[2] Terry Winograd, "Understanding Natural Language," *Cognitive Psychology*, vol. 3, no. 1, pp. 1-191, 1972. [CrossRef] [Google Scholar] [Publisher Link]

[3] Jacob Devlin et al., "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Minneapolis, Minnesota Minneapolis, vol. 1, pp. 4171-4186, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[4] Yinhan Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv preprint*, pp. 1-13, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[5] Kevin Clark et al., "ELECTRA: Pre-Training Text Encoders as Discriminators Rather than Generators," *arXiv preprint*, pp. 1-18, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[6] Pranav Rajpurkar et al., "SQuAD: 100,000+ Questions for Machine Comprehension of Text," *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages*, Austin, Texas, pp. 2383-2392, 2016. [Google Scholar]

[7] Pranav Rajpurkar, Robin Jia, and Percy Liang, "Know What you do not Know: Unanswerable Questions for SQuAD," *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics Melbourne, Melbourne, Australia, vol. 2, pp. 784-789, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[8] Tianyi Zhang et al., "BERTScore: Evaluating Text Generation with BERT," *arXiv preprint*, pp. 1-43, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[9] Thibault Sellam, Dipanjan Das, and Ankur Parikh, "BLEURT: Learning Robust Metrics for Text Generation," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, pp. 7881-7892, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[10] Nils Reimers, and Iryna Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, Association for Computational Linguistics, Hong Kong, China, pp. 3982-3992, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[11] Ricardo Rei et al., "COMET: A Neural Framework for MT Evaluation," *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, pp. 2685-2702, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[12] Kishore Papineni et al., "BLEU: A Method for Automatic Evaluation of Machine Translation," *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, pp. 311-318, 2002. [Google Scholar]

[13] Dang Hoang Long et al., "An Entailment-based Scoring Method for Content Selection in Document Summarization," *Proceedings of the 9th International Symposium on Information and Communication Technology*, Association for Computing Machinery, New York, NY, United States, pp. 122-129, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[14] Rohan Ramanath, Monojit Choudhury, and Kalika Bali, "Entailment: An Effective Metric for Comparing Hierarchical and Non-Hierarchical Annotation Schemes," *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, Sofia, Bulgaria, pp. 42-50, 2013. [Google Scholar]

[15] F. Jelinek et al., "Perplexity-A Measure of the Difficulty of Speech Recognition Tasks," *The journal of the Acoustical Society of America*, vol. 62, no. S1, pp. S63-S63, 1977. [CrossRef] [Google Scholar] [Publisher Link]

[16] Tianyu Gao, Xingcheng Yao, and Danqi Che, "SimCSE: Simple Contrastive Learning of Sentence Embeddings," *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, EMNLP, pp. 6894-6910, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[17] Alexandre Blansché, Pierre Gançarski, and Jerzy J. Korczak, "Genetic Algorithms for Feature Weighting: Evolution vs. Coevolution and Darwin vs. Lamarck," *MICAI 2005: Advances in Artificial Intelligence: 4th Mexican International Conference on Artificial Intelligence*, Monterrey, Mexico, vol. 3789, pp. 682-691, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[18] Md. Monirul Kabir, Md. Shahjahan, and Kazuyuki Murase, "A New Local Search based Hybrid Genetic Algorithm for Feature Selection," *Neurocomputing*, vol. 74, no. 17, pp. 2914-2928, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[19] Benteng Ma, and Yong Xia, "A Tribe Competition-based Genetic Algorithm for Feature Selection in Pattern Classification," *Applied Soft Computing*, vol. 58, pp. 328-338, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[20] Mohammed Ghaith Altarabichi et al., "Fast Genetic Algorithm for Feature Selection-A Qualitative Approximation Approach," *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, Association for Computing Machinery, New York, NY, United States, pp. 11-12, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[21] Abdullah Konak, David W. Coit, and Alice E. Smith, "Multi-Objective Optimization using Genetic Algorithms: A Tutorial," *Reliability Engineering and System Safety*, vol. 91, no. 9, pp. 992-1007, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[22] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014. [CrossRef] [Google Scholar] [Publisher Link]