Reinforcement Learning-based Answer Selection with Class Imbalance Handling and Efficient Differential Evolution Initialization

Jia Wei
College of Education Science, Bohai University
Jinzhou 121000, Liaoning, China

***Abstract*—Answer selection (AS) involves the task of selecting the best answer from a given list of potential options. Current methods commonly approach the AS problem as a binary classification task, using pairs of positive and negative samples. However, the number of negative samples is usually much larger than the positive ones, resulting in a class imbalance. Training on imbalanced data can negatively impact classifier performance. To address this issue, a novel reinforcement learning-based technique is proposed in this study. In this approach, the AS problem is formulated as a sequence of sequential decisions, where an agent classifies each received instance and receives a reward at each step. To handle the class imbalance, the reward assigned to the majority class is lower than that for the minority class. The parameters of the policy are initialized using an improved Differential Evolution (DE) technique. To enhance the efficiency of the DE algorithm, a novel cluster-based mutation operator is introduced. This operator utilizes the K-means clustering approach to identify the winning cluster and employs an upgrade strategy to incorporate potentially viable solutions into the existing population. For word embedding, the DistilBERT model is utilized, which reduces the size of the BERT (Bidirectional encoder representations from transformers) model by 40% and improves computational efficiency by running 60% faster. Despite the decrease, the DistilBERT model maintains 97% of its language comprehension abilities by utilizing knowledge distillation in the pretraining phase. Extensive experiments are carried out on LegalQA, TrecQA, and WikiQA datasets to assess the suggested model. The outcomes showcase the superiority of the proposed model over existing techniques in the domain of AS.**

***Keywords*—Answer selection; imbalanced classification; reinforcement learning; DistilBERT; differential evolution**

I. Introduction

Question Answering (QA) systems, a notable application within natural language processing (NLP) and artificial intelligence (AI), facilitate enhanced human-computer interaction by efficiently processing expansive data and information. Two dominant strategies for developing QA systems include the deployment of Generative Adversarial Networks (GANs) [1] and the utilization of AS techniques. While GANs can generate rich and varied responses, their application comes with challenges related to ensuring grammatical and semantic accuracy in answers. In contrast, AS focuses on meticulously selecting the most apt response from a set of potential answers to a given query, taking into account the inherent variability and complexity of language and potential multiple suitable responses, thereby finding extensive application in various domains including machine comprehension [2]. Both methodologies bring their respective benefits and limitations, influencing their applicability in different use-cases within the broader QA landscape.

Conventional and deep learning techniques offer various methods for AS according to existing literature [3]. While traditional models, like those based on information retrieval, handcrafted rules [4], and machine learning (ML) methods [5] provide certain utilities, they also exhibit limitations in semantic understanding and generalization due to reliance on keywords, manual features, or rigid rules. SVM classifiers within ML approaches have been utilized to connect AS pairs through editing distance and implied matches, yet traditional ML methods often neglect semantic data and demonstrate confined generalizing capacity. Deep learning methods leverage LSTM or CNN architectures to extract semantic features, utilizing their ability to gauge semantic similarity between questions and answers [6]. CNNs model hierarchical sentence structures, while LSTMs ensure representations contain coherent and pertinent information [7]. Notwithstanding their advancements, deep learning models still face challenges in encapsulating comprehensive semantic relationships between questions and answers. To address this, new models, like BERT, harness next-word/phrase prediction and masked word prediction to assimilate complex linguistic relations, outperforming previous models and widely impacting the NLP field [8].

The success of deep algorithms relies heavily on factors like architecture, learning method, and training features, making network design a sophisticated optimization task. Various researchers [9] have addressed this by training neural networks with fixed topologies using several optimization approaches, such as tabu search, ant colony optimization, genetic algorithm, and simulated annealing [10]. Critical to deep models' performance is the optimization of parameter sets, heavily influenced by their initialization [11]. While gradient descent algorithms like Backpropagation (BP) and Levenberg-Marquardt (LM) [12] have been utilized for weight optimization in deep learning methods for AS, their sensitivity to initial weights may lead to local optimum issues. Addressing this, Pretraining weights using Population-based Meta-Heuristic algorithms (PBMH) [13], [14] like DE [15], which incorporates mutation, crossover, and selection steps, has

proven effective for optimizing learning processes by avoiding local minima and ensuring generation of potentially promising solutions [16], [17].

Furthermore, while BERT has established its dominance in NLP tasks due to its deep architecture and ability to capture bidirectional contexts in textual data, its complexity often renders it computationally expensive, especially for real-time applications. Recognizing this challenge, researchers introduced DistilBERT, a distilled version of BERT. The principle behind DistilBERT lies in the concept of knowledge distillation. This technique involves training a smaller model, in this case, DistilBERT, to mimic the behavior and performance of its larger counterpart, BERT. By transferring the knowledge from the cumbersome BERT model to the more lightweight DistilBERT, there is a significant reduction in model size—about 40% smaller than BERT [18]. Notably, despite this reduction, DistilBERT retains a substantial portion of BERT's language comprehension capabilities, making it an efficient alternative for applications demanding both speed and accuracy.

The proposed AS methods utilize binary classifications defined in positive-negative pairs, presenting challenges due to data imbalances as the positive class tends to be smaller than the negative class. This imbalance can degrade model performance but can be addressed through data-level and algorithm-level approaches. Data-level strategies manipulate training data distribution via over/under-sampling of classes, using methods like Synthetic Minority Over-sampling Technique (SMOTE) [19] for creating new samples, and Near Miss [20] for under-sampling by randomly removing samples from the larger class. While under-sampling can omit valuable data, over-sampling might increase over-fitting risk. Algorithmic-level strategies emphasize minority classes through ensemble learning, decision threshold changes, and cost-sensitive learning, which penalizes misclassification of minority class samples. Ensemble learning, in particular, leverages majority voting among multiple classifiers. Additionally, Deep RL has shown promise in various [21] and can manage data imbalance by assigning higher rewards to minority classes in its reward functions [22].

In the realm of AI-driven AS, a pertinent inquiry often raised revolves around the possibility of artificial intelligence (AI) providing a singular, definitive answer. Traditional AS models, by design, sift through potential options to select the most fitting response. However, the rapidly evolving nature of AI and its profound capabilities in understanding intricate data patterns pose a thought-provoking question: can a sophisticated model predict just one conclusive answer, thereby eliminating the need for answer selection? In such a scenario, the fundamental nature of AS would undergo a paradigm shift. The model presented in this paper, with its intricate interplay of RL, DistilBERT word embedding, and enhanced DE, is primarily designed to make the most informed choice from a range of potential answers. While our model showcases efficacy in the AS paradigm, it's worth considering its adaptability in a landscape where AI's aim shifts towards forecasting a singular, precise answer. This perspective not only paves the way for further enhancements to the existing AS models but also encourages a rethinking of AI role in QA systems.

The work introduces an AS model that integrates RL, DistilBERT word embedding, and an enhanced DE method. The model employs two attention-mechanism-based LSTM networks and a feed-forward network, focusing on learning both positive and negative question-answer pairs, utilizing DistilBERT for semantic matching without pre-engineered features. An improved DE algorithm navigates the search space to apply BP algorithms in LSTMs and feed-forward networks, using a selective mutation operator and a novel updating strategy to generate candidate solutions. RL addresses data imbalance in the BP step by treating as a sequential decision-making process. The agent uses environment states for training examples to classify and earn rewards based on correct/incorrect classifications, favoring minority groups in the reward system. The efficacy of the method is demonstrated on three benchmark datasets: TrecQA, LegalQA, and WikiQA, showing superiority over existing models.

Our primary contributions are as follows:

- The adoption of DistilBERT, the state-of-the-art language representation model, for the purpose of attaining sophisticated word embedding, which aims to enrich the semantic understanding of financial texts.
- The introduction of a novel model grounded in RL designed specifically to navigate and mitigate the challenges presented by data imbalance, thereby enhancing the reliability and robustness of the analysis.
- The deployment of an advanced DE algorithm for the crucial task of weight initialization, which is anticipated to augment the predictive accuracy and computational efficiency of the proposed model.

The remaining sections of this article are organized as follows. In Section II, a summary of the relevant work is provided; in Section III, the required background is presented; in Section IV, the structure of the proposed model is described; and in Section V, evaluation metrics, data sets, and results are provided. In Section VI, the study concludes by detailing the lessons learnt and suggesting further work.

II. Related Work

The early studies on AS marked the initial attempts to tackle the task using feature engineering techniques. These methods, such as counting common words, Bag-of-phrases, and Bag-of-grams, provided a basic understanding of the structure and content of questions and answers [23]. However, their reliance on surface-level features limited their ability to capture the deeper semantic nuances inherent in natural language. Recognizing the need to overcome this limitation, subsequent research endeavors delved into more sophisticated approaches for AS. Linguistic tools like WordNet emerged as valuable resources for incorporating semantic knowledge into the selection process [24]. WordNet enabled researchers to enrich the analysis of questions and answers by considering the meanings and associations conveyed by individual words.

Furthermore, researchers sought to exploit the syntactic structure of sentences to enhance AS performance. Techniques like dependency tree analysis and tree distance processing algorithms were employed to capture the relationships between

words and their syntactic roles within a sentence. By considering the hierarchical structure and dependencies encoded in these linguistic representations, researchers aimed to gain deeper insights into the meaning and coherence of questions and answers, enabling more effective selection algorithms. The incorporation of semantic and syntactic analysis in AS research represented a significant shift towards a more comprehensive understanding of language. These approaches recognized that the success of answer selection lies not only in surface-level matching but also in capturing the underlying meaning and context conveyed by questions and answers. As a result, the field witnessed the emergence of more sophisticated methods that combined linguistic tools, syntactic analysis, and semantic knowledge to improve the accuracy and relevance of answer selection algorithms.

In recent years, deep learning models have emerged as powerful tools for AS, leveraging automated feature extraction capabilities to improve performance and enhance the understanding of question-answer pairs [25], [26]. When searching using question-answer pairs, researchers have explored two main approaches. The first approach involves calculating distinct elements in the Q&A pair, with deep networks generating independent representation vectors for questions and answers. To measure the interdependence between these vectors, various criteria have been employed, enabling the comparison and similarity assessment of question-answer pairs [25], [26]. For instance, Wang and Jiang proposed a comparative model that incorporates multiple indicators to measure similarity, taking into account different aspects of the question-answer relationship [25]. Similarly, Yun et al. showcased the advantages of language-based models, utilizing the language model Elmo to capture contextual information and semantic meaning in the question-answer pairs [26]. The second approach treats the query and answer as standalone sentences, allowing researchers to employ specific techniques for their analysis. Severin and Moschitti utilized CNNs to assess the similarity between question-answer pairs, exploiting the local dependencies and patterns within the sentences [27]. On the other hand, Van and Newberg utilized bidirectional LSTM networks, which consider the embedding of words in both directions to capture the contextual information of the question and answer [28]. The resulting relation between the answer and the question is fed into a feed-forward network for further processing and classification. Siamese Networks have also gained popularity in QA tasks, providing separate representation vectors for questions and answers [29], [30]. These networks enable the comparison of the similarity or dissimilarity between question-answer pairs by computing the distance or similarity metrics in the learned feature space. For instance, Yu et al. proposed a deep learning model for AS tasks, employing CNNs and logistic regression to capture the relevant features for answer selection [29]. Similarly, Dryer et al. implemented a similar approach using CNNs and distributed vector representations, enabling the model to learn more nuanced features for question-and-answer representation [30]. To further enhance candidate response selection, researchers have explored pre-processing operations. One such operation involves fixing named entities with unique tokens, simplifying the selection process and enabling better identification of potentially correct answers [31], [32]. This pre-processing step can help alleviate the challenges posed by named entities and improve the accuracy of answer selection. In addition to pre-processing, attention mechanisms have emerged as a valuable strategy in AS research. Initially introduced for machine translation, attention methods have found applications in QA tasks [33]–[35]. These mechanisms allow the model to focus on the most relevant parts of the question and answer by considering the contextual interplay between them. Researchers, such as Jan et al., have proposed using Recurrent Neural Networks (RNNs) with attention mechanisms for response selection, effectively capturing the informative parts of the question-answer pairs [33]. Tay et al. suggested bidirectional alignment and a generalized method based on RNNs further to improve the attention mechanism's performance [34]. Additionally, He et al. demonstrated that combining CNNs with attentional mechanisms can lead to improved performance compared to using RNNs alone [35]. Knowledge-based approaches have also been explored, aiming to leverage external knowledge sources to enhance answer selection. Shen et al. developed a knowledge-based approach that utilizes an attentive bidirectional LSTM network combined with a knowledge graph (KG) to represent questions and answers, enabling the model to leverage structured knowledge to enhance the understanding and selection process [36]. Other techniques have addressed specific challenges in AS, such as data imbalance. Researchers have utilized separate LSTM networks for questions and answers, followed by a Multi-Layer Perceptron (MLP) network for classification, and incorporated per-class penalties to tackle data imbalance and improve classification performance [37]. Matsubara et al. utilized a search engine and a transformer model to select the correct answer, employing models like Jaccard Similarity and Compare Aggregate to assess the relevance of question responses [38]. Furthermore, Kim et al. proposed an architecture based on proximity reference, using an attention mechanism to retain information and automatic encoders better to reduce the information volume, enhancing the model's efficiency and effectiveness [39]. The recent advancements in deep learning models for AS have showcased the versatility and power of these approaches in capturing the complex relationships and semantics present in question-answer pairs. By leveraging techniques such as attention mechanisms, linguistic tools, knowledge graphs, and pre-processing operations, researchers have made significant strides in improving AS performance, ultimately enabling more accurate and relevant answer selection.

Despite the advantages of automatic feature extraction in deep models for AS, there are still several challenges that affect their performance. Typically, these models employ random weight initialization and are trained using the backpropagation BP algorithm to avoid local optima. However, they face difficulties in learning binary classification tasks, particularly when dealing with imbalanced datasets in the context of AS.

III. BACKGROUND

In this section, the prerequisites required to study the rest of the paper are briefly reviewed.

A. Long Short-Term Memory (LSTM)

The LSTM framework, initially brought forward by Hochreiter and Schmidhuber [40], signifies a category of neural structure specifically formulated to proficiently manage the interrelationships within a chain of elements that don't possess a fixed length. The innovative structural design makes it distinctive from conventional neural structures by integrating a storage component within its concealed layer, granting it the capability to comprehend relationships within chains that extend beyond immediate surroundings. This feature equips LSTMs with a particular competence in modelling and interpreting extended dependencies. At the heart of the LSTM architecture is storage elements designed to retain and modify data over a period. This storage unit is made of three critical constituents, often referred to as controllers: the ingress controller (i_t), the oblivion controller (f_t), and the egress controller (o_t). These controllers manage the stream of data within the LSTM unit, facilitating accurate regulation of what data is conserved, disregarded, and exported. The ingress controller (i_t) establishes the extent to which fresh data is incorporated into the storage unit. It considers the present ingress (x_t) and the preceding state of the storage unit (h_{t-1}), and grounded on their interaction, resolves which data is pertinent to refresh the unit state. The oblivion controller oversees the volume of data preserved in the storage unit from the preceding moment. It assesses the current ingress and the preceding storage unit state and resolves what data should be forgotten or discarded from the unit. The egress controller establishes the volume of data from the storage unit that is passed to the egress and influences the concealed state of the LSTM. The egress controller considers the current ingress and the refreshed storage unit state and decides what data should be communicated as the egress. Through the amalgamation of these controllers, the LSTM network can selectively preserve and refresh data over time, equipping it to comprehend both short-term and extended dependencies within sequences. This ability to comprehend and retain pertinent data at appropriate time steps makes LSTMs remarkably competent in an array of assignments such as linguistic processing, speech recognition, and time series prediction.

Mathematically, the LSTM equations can be defined as follows:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (2)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_j x_t + U_j h_{t-1} + b_j) \quad (3)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (4)$$

$$h_t = o_t \tanh(c_t) \quad (5)$$

A bidirectional LSTM (BLSTM) extends an LSTM network to process input from both sides. This can be useful in AS since the answer may be generated by moving the words in the question. In a BLSTM network, the state vectors $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ are generated by parsing the input and combining them as $h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$. LSTM and BLSTM networks treat all the input samples equally important, leading to network confusion. To cope with this problem, an attention mechanism can be considered. To this end, each state h_t is accompanied by the coefficient α_t so the final state h for a sequence of length T is computed as:

$$h_t = \sum_{t=1}^{T} \alpha_t h_t \quad (6)$$

B. Differential Evolution

Differential evolution (DE) [41] has gained widespread recognition as a powerful population-based method capable of effectively solving a wide range of optimization problems [42] DE operates through three essential operations: mutation, crossover, and selection. The DE algorithm commences by initializing a population, usually obtained by sampling from a uniform distribution. This population serves as the foundation for the subsequent evolutionary process. The mutation operator plays a pivotal role in DE, as it generates a mutation vector that introduces diversity and exploration into the population. Through the mutation process, new candidate solutions are created by perturbing the existing individuals in the population. This perturbation is achieved by combining the information from multiple individuals and forming a new candidate solution, often through vector arithmetic operations. The mutation operator in DE typically involves randomly selecting a set of individuals from the population and using their information to compute the mutation vector. This is accomplished by multiplying the difference between two randomly selected individuals by a scaling factor and adding it to a base individual. The resulting mutation vector represents a potential new solution that explores the search space in an attempt to discover better regions of the optimization landscape. The mutation operator in DE plays a crucial role in maintaining population diversity and facilitating exploration. By introducing novel solutions, DE can effectively navigate the optimization landscape and overcome local optima. The quality and diversity of the mutation vector greatly influence the overall performance of DE and its ability to converge to an optimal solution.

The following is the mutation operator that creates a mutation vector:

$$\vec{v}_{i,g} = \vec{x}_{r_1,g} + F\,(\vec{x}_{r_2,g} - \vec{x}_{r_3,g}) \quad (7)$$

where, $\vec{x}_{r_1,g}$, $\vec{x}_{r_2,g}$ and $\vec{x}_{r_3}$ three distinctive candidate solutions are randomly chosen from the current population, and F is a scale factor.

Mutant and target vectors are combined during the crossover. This can be done using the well-known Binomial crossover:

$$u_{i,j,g} = \begin{cases} v_{i,j,g} \text{ if } \mathrm{rand}(0,1) \le \mathrm{CR} \text{ or } \mathrm{j} = j_{rand} \\ x_{i,j,g} \qquad \text{otherwise} \end{cases} \quad (8)$$

where, CR is the crossover ratio, j_{rand} is a random number selected from $\{1,2,\ldots,D\}$ and D is the dimensionality of a candidate solution. After performing crossover, the selection operator selects the target and trial vectors' best solution.

IV. Proposed Model

Fig. 1 depicts the general framework of the suggested technique. Pre-processing, word embedding, and prediction are the three key stages of the proposed technique. As a

preliminary stage, unnecessary words and symbols are eliminated. Using DistilBERT, the embedding vector of each word is retrieved in the second stage, and the similarity between the two sentences is predicted. The suggested method employs a clustering-based differential evolution technique to determine the initial seeds of the network weights, while the RL-based algorithm is used to address the class imbalance.

A. Pre-Processing

Pre-processing is a vital part of any NLP system because the essential characters, words, and sentences identified in this stage are passed to the later stages. Therefore, the pre-processing output has a significant impact on the quality of the final results.

Common stop-word elimination and stemming techniques are employed in the approach. Stop words are part of sentences that can be regarded as overhead. The most common stop words are articles, prepositions, pronouns, etc. They should thus be removed as they cannot function as keywords. For decreasing the dimensionality of the term space, stemming is used to identify the stem of a word. For instance, the terms 'go', 'went', 'going', 'watcher', etc., all can be stemmed from the word "watch". Stemming removes ambiguity and reduces the number of words, time and memory requirements.

B. Word Embedding

Word embedding is used in deep learning algorithms to compare words with semantic vectors. The best technique to produce accurate context-based representations of highlighted words is to insert words.

Many experiments determine the most effective approach to represent words in neural network models. Recently, predefined language models (PLM), previous natural language information boxes, and tuning have been widely used for NLP activities. PLM models frequently use unlabeled data to learn about the model's parameters.

In this article, DistilBERT is considered as one of the newer methods of the PLM model for word input. DistilBERT is an interactive language model designed on large data sets, such as Wikipedia, in order to produce contextual representations. It is common practice to fine-tune the linear layers of DistilBERT for addressing different classification tasks. Some configuration tools teach classification tasks by extracting semantics from common semantic problems or contexts. Models other than DistilBERT build one-directional embeddings which ignore contextual differences. On the contrary, DistilBERT utilizes a bidirectional transformer by conditioning its representations on the left and right context simultaneously.

C. Prediction

Our predictive model comprises two attention-based BLSTMs and one feed-forward network. The two BLSTMs extract embeddings for the question and response sentences. The feed-forward network predicts the degree to which two sentences are similar. Consider $Q = (w_1, w_2, \ldots, w_n)$ and $A = (v_1, v_2, \ldots, v_m)$, where w_i and v_i represent the i-th word in the question and response, respectively.

Because of the length restriction in BLSTM, Q and A can include only n and m words, respectively (in this work, $n = m$). After feeding Q and A into their respective BLSTMs, the attention mechanism computes their embeddings in the following manner:

$$q = \sum_{i=1}^{n} \alpha_i \, h_{q_i} \quad (9)$$

$$r = \sum_{i=1}^{m} \beta_i \, h_{r_i} \quad (10)$$

where, $h_{q_i} = [\overleftarrow{h}_{q_i}, \overrightarrow{h}_{q_i}]$ and $h_{r_i} = [\overleftarrow{h}_{r_i}, \overrightarrow{h}_{r_i}]$ represent the i-th hidden vectors in the BLSTM, and $\alpha_i, \beta_i \in [0,1]$ are the i-th attention weights for each unit in the BLSTM, calculated as:

$$\alpha_i = \frac{e^{u_i}}{\sum_{j=1}^{n} e^{u_j}} \quad (11)$$

$$\beta_i = \frac{e^{v_i}}{\sum_{j=1}^{m} e^{v_j}} \quad (12)$$

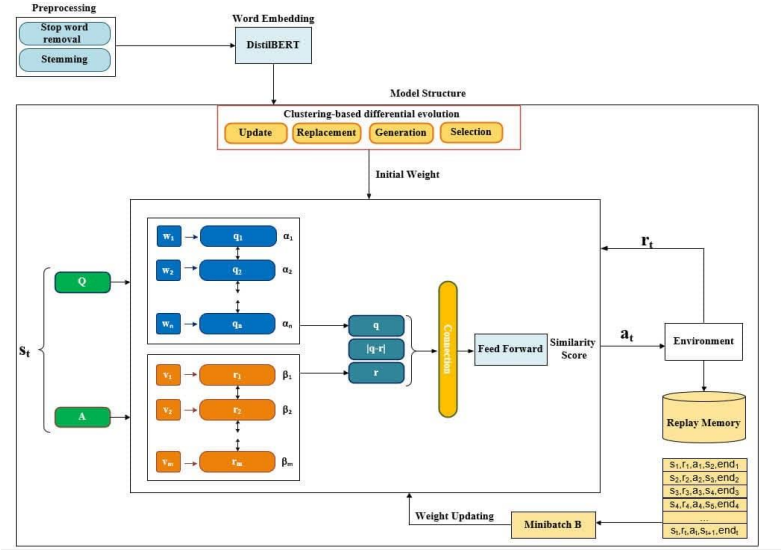

Fig. 1. Overall structure of the proposed model.

$$u_i = tanh(W_u h_{sou_i} + b_u) \quad (13)$$

$$v_i = tanh(W_v h_{sus_i} + b_v) \quad (14)$$

where, W_u, W_v, b_u and b_v are the trainable parameters. As shown in Fig 1, the input of the feed-forward network is the connection of q, r, and $|q - r|$. The training dataset comprises pairs of positive and negative values. Each positive pair comprises a question and its proper response. Each pair of negatives comprises a question and an improper response. Two training phases comprise the model: pretraining and fine-tuning. During pretraining, the augmented differential evolution algorithm is used to determine the optimal initial weights. The initial weights for the fine-tuning phase are the weights obtained during the pretraining phase.

1) Pretraining: The weights of the LSTM, the attention mechanism, and the feed-forward neural network are initialized at this stage. To achieve this, an improved differential evolution method is introduced, incorporating a clustering scheme and a novel fitness function. A clustering-based mutation and update technique is used in the changed DE algorithm to boost the optimization efficiency.

A promising region of the search space is distinguished by the suggested mutation operator, which was inspired by [40]. The k-means clustering algorithm does this by dividing the current set *P* into *k* clusters, each representing a distinct region of the search space. The number of clusters was picked at random from $[2, \sqrt{N}]$. The cluster with the lowest sample means the fit is selected as the optimal group.

The proposed clustering-based mutation is defined as:

$$\overrightarrow{v^{clu}}_i = \overrightarrow{win}_g + F\,(\vec{x}_{r_1,g} - \vec{x}_{r_2,g}) \quad (15)$$

where, $\overrightarrow{win}_g$ is the most acceptable solution in the promising region, and $\vec{x}_{r_1,g}$ and $\vec{x}_{r_2,g}$ are two randomly determined candidate solutions from the current population. It should be noted that $\overrightarrow{win}_g$ is not always the population's most acceptable solution. The clustering-based mutation procedure is implemented *M* times.

The current population is updated when *M* new solutions have been provoked through clustering-based mutation. The steps are as follows:

- Selection: Generate k individuals randomly as initial seeds of the *k*-means algorithm;
- Generation: Generate M solutions using clustering-based mutation as set v^{clu};
- Replacement: Choose M solutions at random and determine as B;
- Update: The best M solutions from the $v^{clu} \cup B$ determined as the B'. The new population is afterwards calculated as $(P - B) \cup B'$.

The fundamental structure of the proposed model comprises two LSTM networks with their respective attention mechanisms and a feed-forward network. As depicted in Fig. 2, in the proposed DE algorithm, all weights and bias terms are organized into a vector to generate a candidate solution.

To assess the quality of a candidate solution, the fitness function is defined as:

$$Fitness = \frac{1}{\sum_{i=1}^{N}(y_i - \tilde{y}_i)^2} \quad (16)$$

where, T is the total number of training samples, y_i is the i-th desired target, and $\tilde{y}_i$ is model prediction.

2) Classification: An RL-based algorithm is employed to tackle the imbalance problem caused by varying data volumes in the classes. Each question-and-answer pair in the training dataset makes up a state of the environment, and the network is the agent that performs a sequence of classifications on all pairs. When the agent predicts the class label of a pair, it is taking an action: the pair seen at the t^{th} time-step is the state s_t, and the classification performed is a_t. In return, the environment provides a reward, r_t, to guide the agent. Reward values are assigned such that classifying a sample from the majority class garners a lower absolute value compared to the minority class. The reward function is:

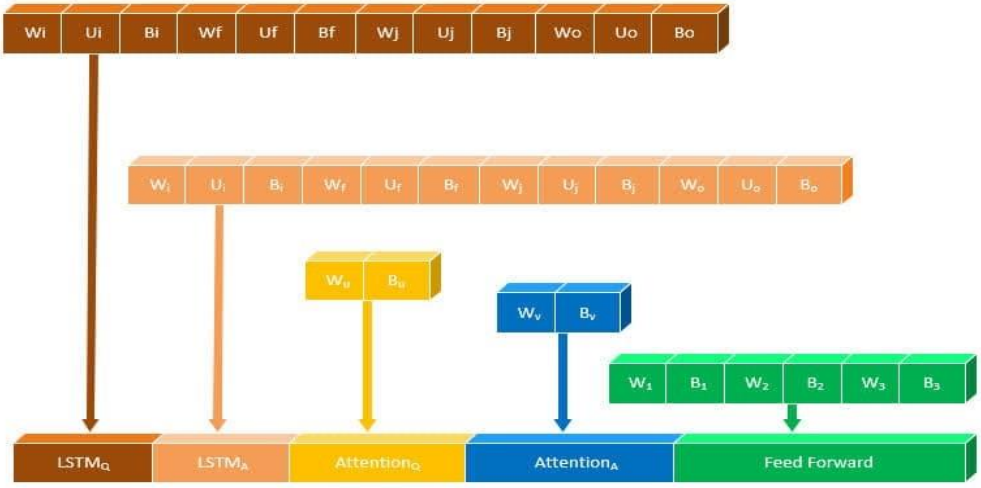

Fig. 2. Encoding strategy in the proposed algorithm.

$$(s_t, a_t, l_t) = \begin{cases} +1\,, a_t = l_t \; and \; s_t \in D_P \\ -1\,, a_t \neq l_t \; and \; s_t \in D_P \\ \lambda\,, a_t = l_t \; and \; s_t \in D_N \\ -\lambda\,, a_t \neq l_t \; and \; s_t \in D_N \end{cases} \quad (17)$$

where, D_P and D_N are the means of the minority and majority classes, respectively. Correct/incorrect classification of a sample from the majority class yields a reward of $+\lambda/-\lambda$, where $0 < \lambda < 1$.

The agent's objective in deep Q-learning is action selection such that the sum of discounted future rewards (R_t) are maximized:

$$R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'} \quad (18)$$

where, γ is the discount factor, $r_{t'}$ is the immediate reward at time step t', and T is the last time-step of the episode. Using γ, more importance is given to rewards in the near future (closer to the current time step t) compared to the distant future. Each episode is terminated if all of the samples are classified correctly or at least one sample from the minority class is misclassified. The expected return of taking action a in state s at time step t and following policy π afterwards is computed as:

$$Q^{\pi}(s,a) = E[R_t | s_t = s, a_t = a, \pi] \quad (19)$$

where, $Q^{\pi}(s,a)$ is called the action-value function. At each state s, the optimal action is the one that maximizes the action-value function:

$$Q^{*}(s,a) = max_{\pi}\, E[R_t | s_t = s, a_t = a, \pi] \quad (20)$$

where, maximization is taken over all possible policies, the recursive form of equation 20 can be written as:

$$Q^{*}(s,a) = E[r + \gamma\, max_{a'} Q^{*}(s', a') | s_t = s, a_t = a] \quad (21)$$

The best action-value function can be estimated iteratively using the Bellman equation:

$$Q_{i+1}(s,a) = E[r + \gamma\, max_{a'} Q_i(s', a') | s_t = s, a_t = a] \quad (22)$$

During training, upon observing state s, the policy network outputs action a. After executing this action, the environment returns a reward r, and the next state becomes s'. The tuple (s, a, r, s') is then saved into the replay memory M. Minibatches B of these tuples are drawn randomly from the replay memory, which is used to update the network parameters via gradient descent. The update is done based on the following loss function:

$$L_i(\theta_i) = \sum_{(s,a,r,s') \in B} (y - Q(s,a;\theta_i))^2 \quad (23)$$

where, θ_i is the network parameters at i-th training iteration, and y is the estimated target for the Q function. The desired target y is equal to the immediate reward for the state-action pair plus the discounted maximum future Q value:

$$y = r + \gamma\, max_{a'} Q(s', a';\, \theta_{k-1}) \quad (24)$$

For terminal states, y is equal to r. At *ith* iteration, the gradient of the loss function is calculated as follows:

$$\nabla_{\theta_i} L(\theta_i) = -2 \sum_{(s,a,r,s') \in B} (y - Q(s,a;\theta_i)) \nabla_{\theta_i} Q(s,a;\theta_i) \quad (25)$$

The network weights are updated using the gradient of loss function computed as follows:

$$\theta_{i+1} = \theta_i + \alpha \nabla_{\theta_i} Q(s,a;\theta_i) \quad (26)$$

where, α is the learning rate.

V. Experimental Results

In this section, the conducted experiments are detailed.

A. Datasets

The following three benchmark databases are used during the experiments (see Table I for their statistics):

- TrecQA [43] is taken from the TREC trace dataset. Yao et al. [44] used two training datasets, TRAIN, and TRAIN-ALL, to construct an extended set of positive and negative pairs. The soundness of answers in the TRAIN-ALL dataset is verified automatically by matching pairs with regular expressions. The TRAIN, DEV, and TEST data set' responses were all manually assessed. To teach the model, the TRAIN-ALL set is utilized.
- LegalQA [45] is a database of legal question-and-answer submissions from the Chinese community. Inquiries were answered online by a licensed attorney. The four fields that make up LegalQA are Question Title, Question Text, Answer, and Label. A straight line designates real positive couples.
- A Wikipedia page that is regarded as a subject of the year is linked to each question in the open-source quality assurance dataset known as WikiQA. [46]. To avoid ambiguity in the answer sentences, all the answers at the bottom of the page are the candidates' answers.

B. Metrics

According to earlier research, the most popular reference points for the answer-selection task are MAP and MRR [47]. MAP evaluates the capacity to categorize responses and return solutions. If a high score match is found, the MRR is repeated. The average accuracy is derived using the Mean Average Precision (MAP) findings:

TABLE I. Statistical Information from the LegalQA, TrecQA and WikiQA Datasets

	Question			QA pairs			Correct		
dataset	***train***	***dev***	***test***	***train***	***dev***	***test***	***train dev test***		
LegalQA	10,526	1,593	3,035	100,590	11,965	26,913	21.8	24.4	22.9
TracQA	1,229	82	100	53,417	1,148	1,517	12.0	19.3	18.7
WikiQA	873	126	243	20,360	1,130	2,352	12.0	12.4	12.5

$$MAP(Q)=\frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{n_i}\sum_{j=1}^{n_i} Precision\ (R_{ij}) \quad (27)$$

where, Q is the questions set, n_i is the number of relevant answers to i-th question, and R_{ij} is the set of j best candidates selected from the n_i available answers. The position of the first correct response is used to determine the Mean Reciprocal Rank (MRR) calculated as follows:

$$\text{MRR (Q)} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{r_i} \quad (28)$$

where, r_i denotes the first response's placement for i-th question. Details of model

The experiments were implemented using Python and PyTorch. For natural language processing in Python, the NLTK package was leveraged. A two-layer LSTM with a hidden size of 64 was employed. Also, because there are connections between the vectors in the two LSTM networks, the batch must be normalized before it is sent to the feed-forward neural network. The tests were conducted using a computer with 64 GB Memory, a 64-bit Windows operating system, and a graphics processing unit (GPU). The most effective models for LegalQA, TrecQA, and WikiQA were identified after 50, 60, and 100 epochs. For the three datasets, training took 5, 20, and 60 hours, respectively.

C. Model Performance

First, the system was tested against nine deep learning-based strategies, including KABLSTM [48], EATS [49], AM-BLSTM [50], BERT-Base [51], DRCN [52], P-CNN [53], DARCNN [54], DASL [55], IKAAS [56]. The outcomes for the three datasets—LegalQA, TrecQA, and WikiQA—are presented in Table II. All the trials were carried out five times to avoid the randomness of heuristic algorithms influencing the findings. Results using random weight initialization (Proposed (no RL and DE)), enhanced DE (Proposed (no RL)), RL use (Proposed (no DE)), and the entire model are shown for the suggested method (Proposed).

Table III displays the extent to which the proposed method outperforms other methods. The proposed model consistently demonstrates a significant advantage over other widely-recognized methods in the domain. When examining the MRR and MAP metrics specifically for the LegalQA dataset, the proposed model exhibits enhancements ranging from +0.077 to +0.231, with the most pronounced improvement observed against the DARCNN method. This consistent performance is evident across all datasets, underscoring the robustness and adaptability of the proposed approach. Notably, even the variants of the proposed model, such as "Proposed (no RL and DE)" and "Proposed (no RL)", consistently outpace most other techniques. These modified versions, despite lacking certain features, still deliver commendable results, emphasizing the intrinsic strength of the primary model. An intriguing point is the comparison between the BERT-Base and its more streamlined version, DistilBERT. While BERT-Base stands as a powerful model in the NLP realm, the margin table shows that the approach of the proposed model surpasses it, attesting to the innovative methods integrated into the new model. Addressing Imbalance: The performances of models like P-CNN and DARCNN, especially the substantial gains in specific metrics such as +0.285 for TrecQA (MRR) and +0.231 for LegalQA (MRR), shed light on the challenges presented by data imbalance in the AS domain. The resilience and adaptability of the proposed model to such challenges, coupled with its ability to deliver top-notch results, underscore its potential in addressing imbalanced datasets effectively.

TABLE II. PERFORMANCE COMPARISON OF THE PROPOSED MODEL WITH THOSE ALREADY IN USE ON THREE DATASETS: RESULTS USING THE DAG MARKER WERE FOUND IN EARLIER STUDIES

Method	LegalQA		TrecQA		WikiQA	
	MRR	***MAP***	***MRR***	***MAP***	***MRR***	***MAP***
KABLSTM	0.752	0.784	0.792†	0.844†	0.732†	0.749†
EATS	0.780	0.838	0.854†	0.881†	0.700†	0.715†
AM-BLSTM	0.787	0.814	0.806	0.842	0.843	0.794
BERT-Base	0.830	0.841	0.837	0.831	0.816†	0.828†
DRCN	0.856	0.846	0.823	0.846	0.804†	0.862†
P-CNN	0.735	0.742	0.673	0.714	0.734†	0.737†
DARCNN	0.708	0.752	0.765	0.748	0.734†	0.750†
DASL	0.821	0.815	0.846	0.848	0.781	0.778
IKAAS	0.825†	0.883†	0.823†	0.868†	0.846	0.845
Proposed (no RL and DE)	0.742 ± 0.017	0.826 ± 0.005	0.791 ± 0.014	0.825 ± 0.025	0.759 ± 0.017	0.732 ± 0.015
Proposed (no RL)	0.796 ± 0.021	0.841 ± 0.019	0.831 ± 0.026	0.856 ± 0.026	0.831 ± 0.014	0.843 ± 0.048
Proposed (no DE)	0.862 ± 0.019	0.859 ± 0.031	0.858 ± 0.019	0.874 ± 0.016	0.849 ± 0.012	0.856 ± 0.010
Proposed	0.939 ± 0.018	0.955 ± 0.096	0.958 ± 0.039	0.941 ± 0.085	0.912 ± 0.035	0.929 ± 0.031

TABLE III. MARGIN OF IMPROVEMENT OF THE PROPOSED MODEL OVER OTHER METHODS

Method	LegalQA		TrecQA		WikiQA	
	MRR	***MAP***	***MRR***	***MAP***	***MRR***	***MAP***
KABLSTM	+0.187	+0.171	+0.166	+0.097	+0.180	+0.180
EATS	+0.159	+0.117	+0.104	+0.060	+0.212	+0.214
AM-BLSTM	+0.152	+0.141	+0.152	+0.099	+0.069	+0.135
BERT-Base	+0.109	+0.114	+0.121	+0.110	+0.096	+0.101
DRCN	+0.083	+0.109	+0.135	+0.095	+0.108	+0.067
P-CNN	+0.204	+0.213	+0.285	+0.227	+0.178	+0.192
DARCNN	+0.231	+0.203	+0.193	+0.193	+0.178	+0.179
DASL	+0.118	+0.140	+0.112	+0.093	+0.131	+0.151
IKAAS	+0.114	+0.072	+0.135	+0.073	+0.066	+0.084
Proposed (no RL and DE)	+0.197	+0.129	+0.167	+0.116	+0.153	+0.197
Proposed (no RL)	+0.143	+0.114	+0.127	+0.085	+0.081	+0.086
Proposed (no DE)	+0.077	+0.096	+0.100	+0.067	+0.063	+0.073

1) Comparison with other metaheuristics: In this section, a variety of meta-heuristic optimization algorithms are compared to the enhanced DE algorithm. To do this, a variety of meta-heuristics are employed while maintaining the integrity of the other model elements, such as pre-processing, word embedding, LSTM, network structure, and RL, in order to gain the initial model parameters. Eight different algorithms, including (standard) DE [57], grey wolf optimization (GWO) [58], bat algorithm (BA) [59], dragonfly algorithm (DA) [47], salp swarm algorithm (SSA) [60], cuckoo optimization algorithm (COA) [61], human mental search (HMS) [40], whale optimization algorithm (WOA) [62], and artificial bee colony (ABC) [63] are investigated.

The overall size of all algorithms and their predicted capacities were calculated to be 150 and 4,000, respectively. In Table IV, the default settings can be observed. Table V displays the findings for each of the three data sets. On every dataset, the suggested DE algorithm performs better than any other algorithm, as shown. Normal DE is the runner-up.

TABLE IV. SETTING PARAMETERS FOR META-HEURISTICS

algorithm	parameter	value
DE	scaling factor	0.4
	crossover probability	0.7
BAT	loudness update constant	0.60
	emission rate update constant	0.50
	initial pulse emission rate	0.001
COA	alien solutions discovery rate	0.25
HMS	Maximum mental processes	5
	C	1
	loudness update constant	0.50
WOA	b	1
ABC	limit	ne× dimensionality
	no	50% of the colony
	ne	50% of the colony
	ns	1

TABLE V. RESULTS OF META-HEURISTIC ALGORITHMS ON THE DATASETS FROM LEGALQA, TRECQA, AND WIKIQA

Method	LegalQA		TrecQA		WikiQA	
	MRR	***MAP***	***MRR***	***MAP***	***MRR***	***MAP***
DE	0.915 ± 0.019	0.933 ± 0.015	0.890 ± 0.046	0.916 ± 0.191	0.872 ± 0.036	0.911 ± 0.009
GWO	0.774 ± 0.116	0.771 ± 0.090	0.741 ± 0.075	0.783 ± 0.134	0.742 ± 0.038	0.771 ± 0.016
BAT	0.855 ± 0.013	0.809 ± 0.028	0.867 ± 0.088	0.874 ± 0.295	0.842 ± 0.071	0.863 ± 0.090
DA	0.809 ± 0.085	0.819 ± 0.039	0.859 ± 0.015	0.876 ± 0.053	0.816 ± 0.090	0.850 ± 0.083
SSA	0.739 ± 0.030	0.756 ± 0.081	0.745 ± 0.082	0.756 ± 0.017	0.739 ± 0.053	0.753 ± 0.023
COA	0.857 ± 0.091	0.883 ± 0.015	0.880 ± 0.073	0.895 ± 0.249	0.870 ± 0.013	0.860 ± 0.019
HMS	0.841 ± 0.010	0.875 ± 0.193	0.875 ± 0.018	0.890 ± 0.047	0.837 ± 0.159	0.863 ± 0.159
WOA	0.752 ± 0.016	0.753 ± 0.027	0.769 ± 0.05	0.789 ± 0.085	0.731 ± 0.000	0.760 ± 0.018
ABC	0.873 ± 0.014	0.896 ± 0.038	0.875 ± 0.025	0.889 ± 0.015	0.872 ± 0.020	0.881 ± 0.010

2) Reward function: The reward function directs the agent toward achieving its aim by giving the right ratings to certain activities. ±1 and ±λ were selected as the rewards for the minority and majority classes, respectively. The ratio of the sample size of the majority class to the minority class determines the λ value. As the majority/minority ratio rises, the λ value decreases. The majority class bonus is held constant, and λ is chosen from the set {0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1} to see how changing λ affects the reward earned by the model. The evaluation findings for the three datasets are displayed in Fig. 3. The reward plots in Fig. 3(a), Fig. 3(b), and Fig. 3(c) all have an ascending trend for $\lambda < 0.4$ and a decreasing trend for $\lambda > 0.4$. The relevance of majority classes is disregarded for $\lambda = 0$, while for $\lambda = 1$, both classes are regarded as equally significant. Even though the minority is more important to us, the impact of the majority should not be ignored.

3) Examples: A qualitative example is provided to evaluate the efficacy of RL in the model, focusing on the question "Who is the president or chief executive of Amtrak?" from the TrecQA dataset. The results of the top five answers retrieved by the model with and without using RL are shown in Fig. 4. As seen, models without RL are more likely to select negative answers. The model with RL gives the highest possible score for answering the question.Word embeddings.

In this section, performing the DistilBERT adopted in the method for word embedding is compared against five other word embedding methods. One-Hot Encoding [64] creates binary properties for each class and assigns values to the properties in each instance that corresponds to a specific class. CBOW and Skip-gram [65] use neural networks to compare words with insertion vectors. GloVe [66] is an unattended learning algorithm implemented for full word set statistics. FastText [67] [65] extends the Skip-gram model, in which each word is represented by an n-gram character instead of learning a vector for words. Table VI shows the results of the conducted experiment. As expected, One-Hot cryptography has the lowest performance among the evaluated methods. CBOW and Skip-gram perform similarly, and both yield better performance compared to GloVe, while FastText gives better results. However, the best performance is claimed by the DistilBERT model, which is the motivation behind its use in the approach.

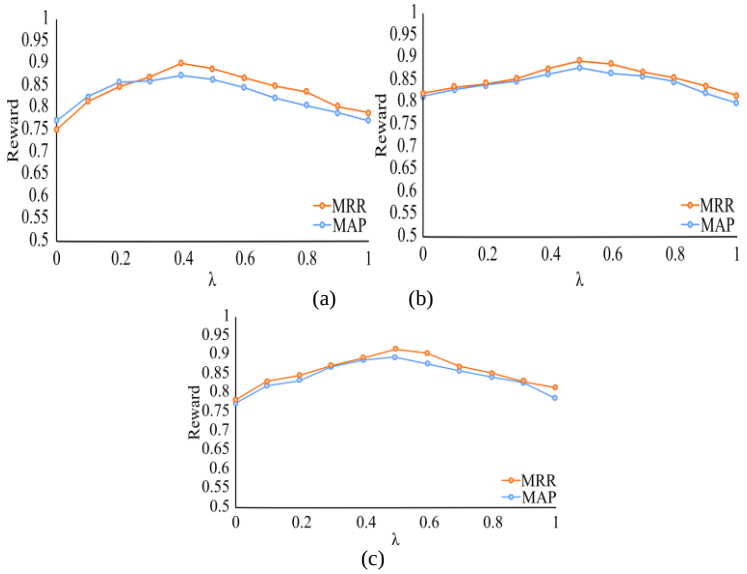

Fig. 3. Reward vs. λ for (a) LegalQA dataset, (b) TrecQA dataset, and (c) WikiQA dataset.

Rank	Ranked answers w/o RL	Ranked answers by RL
1	**george warrington**, amtrak's president, said ridership was up because service was better and because" the marketplace is feeling increasingly frustrated with the alternatives, " namely jammed highways and delayed airplanes.	"long-term success here has to do with doing it right, getting it right and increasing market share, "said **george warrington**, amtrak's president and chief executive.
2	amtrak president **george warrington** said settling out-of-court was the right thing to do.	amtrak president **george warrington** said settling out-of-court was the right thing to do.
3	amtrak is offering a deal it hopes few travelers can resist: get good service or a free ride.	amtrak president **george warrington** told the committee that the railway expects delivery next week of the first of 20 high-speed trains for the boston -to- washington northeast corridor.
4	amtrak spokesman john wolf estimated that only one out of every 1,000 passengers will request a refund voucher.	"amtrak is committed to treating all employees fairly, "amtrak president **george warrington** said in a statement.
5	amtrak president **george warrington** told the committee that the railway expects delivery next week of the first of 20 high-speed trains for the boston-to-washington northeast corridor.	**george warrington**, amtrak's president, said ridership was up because service was better and because" the marketplace is feeling increasingly frustrated with the alternatives," namely jammed highways and delayed airplanes.

Fig. 4. " Who is the president or CEO?" This table shows the top 5 answers for models with and without RL. " George Warrington" is the field answer, and the underlined word refers to that term.

TABLE VI. RESULTS OF DIFFERENT WORD EMBEDDINGS ON THE THREE DATASETS

Word embedding	LegalQA		TrecQA		WikiQA	
	MRR	***MAP***	***MRR***	***MAP***	***MRR***	***MAP***
One Hot encoding	0.679 ± 0.042	0.569 ± 0.002	0.711 ± 0.120	0.653 ± 0.081	0.649 ± 0.089	0.589 ± 0.093
CBOW	0.851 ± 0.027	0.840 ± 0.015	0.880 ± 0.081	0.861 ± 0.126	0.838 ± 0.023	0.820 ± 0.019
Skip-gram	0.874 ± 0.052	0.872 ± 0.075	0.878 ± 0.030	0.858 ± 0.002	0.847 ± 0.014	0.853 ± 0.014
Glove	0.812 ± 0.027	0.853 ± 0.082	0.795 ± 0.140	0.821 ± 0.074	0.782 ± 0.039	0.806 ± 0.009
FastText	0.879 ± 0.012	0.901 ± 0.041	0.886 ± 0.093	0.876 ± 0.002	0.861 ± 0.099	0.870 ± 0.000

4) Discussion: The proposed model in this study addressed the class imbalance issue in AS by employing a reinforcement learning-based technique. Unlike traditional methods that treat it as a binary classification problem, the proposed approach formulated it as a sequence of sequential decisions. An agent classified each instance and received a reward at each step. To handle class imbalance, the reward assigned to the majority class was intentionally lower than that for the minority class. The parameters of the policy were initialized using an improved DE technique. To improve the efficiency of the DE algorithm, a novel cluster-based mutation operator was introduced. This operator utilized the K-means clustering approach to identify the winning cluster and incorporated potentially viable solutions into the existing population. In terms of word embedding, the model employed the DistilBERT model, which reduced the size of the BERT model. To evaluate the effectiveness of the proposed model, extensive experiments were conducted using LegalQA, TrecQA, and WikiQA datasets. The results demonstrated the superiority of the proposed model compared to existing methods in the field of answer selection. However, it is important to acknowledge certain limitations of the proposed model, which can be considered in future work:

a) Limited Scope: While the article introduces a novel reinforcement learning-based technique to address class imbalance in AS, it is important to acknowledge that class imbalance is a widely recognized challenge in machine learning, and various approaches have been proposed in the literature. A more comprehensive discussion that explores alternative methods, such as data resampling techniques (e.g., oversampling, under sampling), cost-sensitive learning, or ensemble-based methods, would provide a broader perspective on addressing the class imbalance in AS tasks. Comparing the proposed technique with these alternative approaches in terms of effectiveness and applicability would enhance the understanding of its limitations and potential alternatives.

b) Lack of Real-World Application: The evaluation of the proposed model on LegalQA, TrecQA, and WikiQA datasets provides insights into its performance within specific domains. However, it is important to recognize that these datasets might not fully capture the complexities and variations present in real-world AS scenarios. To overcome this limitation, future research should consider evaluating the proposed technique on diverse datasets from different domains, such as medical, finance, or customer support, to assess its generalizability and robustness across various real-world applications. This would provide a more comprehensive understanding of the technique's effectiveness and limitations in practical settings.

c) Performance Metrics and Statistical Significance: While the article claims superiority over existing methods, it is essential to provide a detailed analysis of the performance metrics used for evaluation. Precision, recall, F1-score, and other relevant metrics should be reported, along with the corresponding confidence intervals or statistical tests, to establish the statistical significance of the results. A thorough analysis of these metrics would provide a clearer understanding of the proposed technique's performance and its potential limitations in different AS scenarios.

d) Computational Efficiency: While the utilization of the DistilBERT model is mentioned to enhance computational efficiency, it would be beneficial to provide more specific details about the computational resources required by the proposed technique. Comparing the computational requirements, such as memory usage and processing time, with other state-of-the-art AS methods would allow for a more comprehensive assessment. Additionally, considering the scalability of the technique for larger datasets or real-time applications would provide insights into its feasibility and practical utility in various contexts.

e) Interpretability and Explainability: The article lacks discussion on the interpretability and explainability of the proposed technique. In AS tasks, understanding the decision-making process and providing explanations for selected answers are important factors for trust and transparency. Discussing methods or approaches used to interpret and explain the decisions made by the reinforcement learning-based model would enhance its applicability in real-world scenarios. Consideration of techniques like attention mechanisms or post-hoc interpretability methods (e.g., LIME, SHAP) would provide insights into the reasoning behind answer selections and potential biases or limitations associated with the model's decisions.

f) User Feedback and Adaptability: The article does not discuss the potential for incorporating user feedback or adapting the AS system over time. AS models that can learn from user interactions, such as reinforcement learning with online learning or active learning approaches, have the potential to improve their performance based on user preferences and changing information needs. Investigating the integration of user feedback and methods for continuous adaptation would be valuable for enhancing the proposed technique's effectiveness and user satisfaction.

g) Comparison with Human Performance: The article focuses on comparing the proposed model with existing methods, but it does not include a comparison with human

performance. AS tasks often involve subjective judgments, and comparing the performance of the proposed technique with human experts or crowd-sourced annotations can provide valuable insights into the model's strengths and limitations. Conducting experiments that involve human evaluations would help contextualize the performance of the proposed technique and highlight areas where further improvements are needed.

h) Ensuring Data Quality and Model Performance: Another aspect warranting discussion is the challenge of recognizing datasets that may potentially misguide the classifier. Any model's efficiency is contingent upon the quality and reliability of its training data. Datasets that contain noisy, inconsistent, or unrepresentative samples can induce biases in the model, leading to flawed predictions. Regular monitoring of performance metrics on validation sets can provide early indications of a model being misguided by its data. A substantial divergence between training and validation performance may hint towards potential dataset issues. Tools like ChatGPT and other advanced language models can offer benefits in this scenario. These models, with their vast training on diverse textual data, can be harnessed to validate the coherence and authenticity of data samples. For instance, they could generate synthetic samples for augmentation, thereby balancing datasets and mitigating risks. They can also be employed to highlight potential anomalies or inconsistencies within a dataset, aiding in its refinement and preprocessing. In future studies, integrating insights from these tools could be an invaluable step for data validation, ensuring models are trained on high-quality, representative datasets.

VI. Conclusion

In this paper, an approach for efficient AS is proposed, which employs enhanced DE algorithms for pretraining and RL for instructing the BP algorithm. The method is based on LSTM with an attention mechanism and DistilBERT word embedding. The proposed model categorizes both positive and negative classes and comprises pairs of positive inquiries and detailed responses. Because the dataset contains many negative pairs, the proposed model produces an unbalanced classification. To address this issue, the approach is framed as a logical decision-making process. Correct classification of minority samples is rewarded with higher values at each episode step than the correct classification of the majority samples. Each episode is repeated until a minority sample is misclassified or all samples are correctly classified. The policy weights were initialized using an improved DE algorithm. The improved DE algorithm clusters the current population and finds promising regions in the search space using a new upgrade strategy. The evaluation of the proposed method was conducted using the LegalQA, TrecQA, and WikiQA datasets, demonstrating its superior performance compared to other methods.

In addition to the proposed classification approach, there are several promising avenues for future research in the field of Natural Language Processing (NLP). One area of interest is exploring the utility of the proposed approach in various NLP applications beyond answer selection. By applying the same reinforcement learning-based technique to tasks such as sentiment analysis, text summarization, or named entity recognition, Insights into the effectiveness and generalizability across different domains can be gained through the study.

Another promising direction for future research is the provision of candidate answers to given questions. While the proposed approach focuses on selecting the best answer from a given list of options, the generation of candidate answers could further enhance the AS process. One potential approach to generating candidate answers is through the use of Generative Adversarial Networks (GANs). GANs have shown promise in generating realistic and coherent text, and their application in generating diverse and plausible candidate answers could greatly enrich the AS process. Further investigation into the integration of GANs with the proposed classification approach could lead to more comprehensive and accurate answer selection systems.

References

[1] L. Hong et al., "GAN-LSTM-3D: An efficient method for lung tumour 3D reconstruction enhanced by attention-based LSTM," CAAI Trans Intell Technol, 2023.

[2] J. Huang, "A multi-size neural network with attention mechanism for answer selection," arXiv preprint arXiv:2105.03278, 2021.

[3] S. Zhang, X. Zhang, H. Wang, J. Cheng, P. Li, and Z. Ding, "Chinese medical question answer matching using end-to-end character-level multi-scale CNNs," Applied Sciences, vol. 7, no. 8, p. 767, 2017.

[4] X. Xu, F. Shen, Y. Yang, H. T. Shen, and X. Li, "Learning discriminative binary codes for large-scale cross-modal retrieval," IEEE Transactions on Image Processing, vol. 26, no. 5, pp. 2494–2507, 2017.

[5] S.-J. Yen, Y.-C. Wu, J.-C. Yang, Y.-S. Lee, C.-J. Lee, and J.-J. Liu, "A support vector machine-based context-ranking model for question answering," Inf Sci (N Y), vol. 224, pp. 77–87, 2013.

[6] W. Yin, M. Yu, B. Xiang, B. Zhou, and H. Schütze, "Simple question answering by attentive convolutional neural network," arXiv preprint arXiv:1606.03391, 2016.

[7] S. V. Moravvej, A. Mirzaei, and M. Safayani, "Biomedical text summarization using conditional generative adversarial network (CGAN)," arXiv preprint arXiv:2110.11870, 2021.

[8] S. V Moravvej, M. J. Maleki Kahaki, M. Salimi Sartakhti, and M. Joodaki, "Efficient GAN-based method for extractive summarization," Journal of Electrical and Computer Engineering Innovations (JECEI), vol. 10, no. 2, pp. 287–298, 2022.

[9] R. S. Sexton, R. E. Dorsey, and J. D. Johnson, "Optimization of neural networks: A comparative analysis of the genetic algorithm and simulated annealing," Eur J Oper Res, vol. 114, no. 3, pp. 589–601, 1999.

[10] S. Mirjalili and S. Mirjalili, "Genetic algorithm," Evolutionary Algorithms and Neural Networks: Theory and Applications, pp. 43–55, 2019.

[11] C. A. R. de Sousa, "An overview on weight initialization methods for feedforward neural networks," in 2016 International Joint Conference on Neural Networks (IJCNN), IEEE, 2016, pp. 52–59.

[12] A. Ranganathan, "The levenberg-marquardt algorithm," Tutoral on LM algorithm, vol. 11, no. 1, pp. 101–110, 2004.

[13] S. Vakilian, S. V. Moravvej, and A. Fanian, "Using the artificial bee colony (ABC) algorithm in collaboration with the fog nodes in the Internet of Things three-layer architecture," in 2021 29th Iranian Conference on Electrical Engineering (ICEE), IEEE, 2021, pp. 509–513.

[14] S. Vakilian, S. V. Moravvej, and A. Fanian, "Using the cuckoo algorithm to optimizing the response time and energy consumption cost of fog nodes by considering collaboration in the fog layer," in 2021 5th International Conference on Internet of Things and Applications (IoT), IEEE, 2021, pp. 1–5.

[15] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," Journal of global optimization, vol. 11, pp. 341–359, 1997.

[16] S. V. Moravvej, S. J. Mousavirad, M. H. Moghadam, and M. Saadatmand, "An LSTM-based plagiarism detection via attention mechanism and a population-based approach for pre-training parameters with imbalanced classes," in Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part III 28, Springer, 2021, pp. 690–701.

[17] S. Danaei et al., "Myocarditis Diagnosis: A Method using Mutual Learning-Based ABC and Reinforcement Learning," in 2022 IEEE 22nd International Symposium on Computational Intelligence and Informatics and 8th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Science and Robotics (CINTI-MACRo), IEEE, 2022, pp. 265–270.

[18] M. Schütz, A. Schindler, M. Siegel, and K. Nazemi, "Automatic fake news detection with pre-trained transformer models," in Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10-15, 2021, Proceedings, Part VII, Springer, 2021, pp. 627–641.

[19] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning," in International conference on intelligent computing, Springer, 2005, pp. 878–887.

[20] A. R. B. Alamsyah, S. R. Anisa, N. S. Belinda, and A. Setiawan, "Smote and nearmiss methods for disease classification with unbalanced data: Case study: IFLS 5," in Proceedings of The International Conference on Data Science and Official Statistics, 2021, pp. 305–314.

[21] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in 2017 IEEE international conference on robotics and automation (ICRA), IEEE, 2017, pp. 3389–3396.

[22] S. V. Moravvej et al., "RLMD-PA: A reinforcement learning-based myocarditis diagnosis combined with a population-based algorithm for pretraining weights," Contrast Media Mol Imaging, vol. 2022, 2022.

[23] A. Severyn and A. Moschitti, "Automatic feature engineering for answer selection and extraction," in Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 458–467.

[24] S. W. Yih, M.-W. Chang, C. Meek, and A. Pastusiak, "Question answering using enhanced lexical semantic models," in Proceedings of the 51st Annual Meeting of the Association for Computational linguistics, 2013.

[25] S. Wang and J. Jiang, "A compare-aggregate model for matching text sequences," arXiv preprint arXiv:1611.01747, 2016.

[26] S. Yoon, F. Dernoncourt, D. S. Kim, T. Bui, and K. Jung, "A compare-aggregate model with latent clustering for answer selection," in Proceedings of the 28th ACM international conference on information and knowledge management, 2019, pp. 2093–2096.

[27] A. Severyn and A. Moschitti, "Learning to rank short text pairs with convolutional deep neural networks," in Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, 2015, pp. 373–382.

[28] D. Wang and E. Nyberg, "A long short-term memory model for answer sentence selection in question answering," in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), 2015, pp. 707–712.

[29] L. Yu, K. M. Hermann, P. Blunsom, and S. Pulman, "Deep learning for answer sentence selection," arXiv preprint arXiv:1412.1632, 2014.

[30] M. Feng, B. Xiang, M. R. Glass, L. Wang, and B. Zhou, "Applying deep learning to answer selection: A study and an open task," in 2015 IEEE workshop on automatic speech recognition and understanding (ASRU), IEEE, 2015, pp. 813–820.

[31] H. T. Madabushi, M. Lee, and J. Barnden, "Integrating question classification and deep learning for improved answer selection," in Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 3283–3294.

[32] J. Rao, H. He, and J. Lin, "Noise-contrastive estimation for answer selection with deep neural networks," in Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, 2016, pp. 1913–1916.

[33] L. Yang, Q. Ai, J. Guo, and W. B. Croft, "anmm: Ranking short answer texts with attention-based neural matching model," in Proceedings of the 25th ACM international on conference on information and knowledge management, 2016, pp. 287–296.

[34] Y. Tay, L. A. Tuan, and S. C. Hui, "Co-stack residual affinity networks with multi-level attention refinement for matching text sequences," arXiv preprint arXiv:1810.02938, 2018.

[35] H. He, J. Wieting, K. Gimpel, J. Rao, and J. Lin, "UMD-TTIC-UW at SemEval-2016 Task 1: Attention-based multi-perspective convolutional neural networks for textual similarity measurement," in Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016, pp. 1103–1108.

[36] Y. Shen et al., "Knowledge-aware attentive neural network for ranking question answer pairs," in The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018, pp. 901–904.

[37] S. V. Moravvej, M. J. M. Kahaki, M. S. Sartakhti, and A. Mirzaei, "A method based on attention mechanism using bidirectional long-short term memory (BLSTM) for question answering," in 2021 29th Iranian Conference on Electrical Engineering (ICEE), IEEE, 2021, pp. 460–464.

[38] Y. Matsubara, T. Vu, and A. Moschitti, "Reranking for efficient transformer-based answer selection," in Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, 2020, pp. 1577–1580.

[39] S. Kim, I. Kang, and N. Kwak, "Semantic sentence matching with densely-connected recurrent and co-attentive information," in Proceedings of the AAAI conference on artificial intelligence, 2019, pp. 6586–6593.

[40] S. J. Mousavirad and H. Ebrahimpour-Komleh, "Human mental search: a new population-based metaheuristic optimization algorithm," Applied Intelligence, vol. 47, pp. 850–887, 2017.

[41] S. V. Moravvej, S. J. Mousavirad, D. Oliva, G. Schaefer, and Z. Sobhaninia, "An improved de algorithm to optimise the learning process of a bert-based plagiarism detection model," in 2022 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2022, pp. 1–7.

[42] S. V. Moravvej, S. J. Mousavirad, D. Oliva, and F. Mohammadi, "A Novel Plagiarism Detection Approach Combining BERT-based Word Embedding, Attention-based LSTMs and an Improved Differential Evolution Algorithm," arXiv preprint arXiv:2305.02374, 2023.

[43] E. M. Voorhees, "The evaluation of question answering systems: Lessons learned from the TREC QA track," in Question Answering: Strategy and Resources Workshop Program, Citeseer, 2002, p. 6.

[44] X. Yao, B. Van Durme, C. Callison-Burch, and P. Clark, "Answer extraction as sequence tagging with tree edit distance," in Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies, 2013, pp. 858–867.

[45] W. Huang, J. Jiang, Q. Qu, and M. Yang, "AILA: A Question Answering System in the Legal Domain.," in IJCAI, 2020, pp. 5258–5260.

[46] Y. Yang, W. Yih, and C. Meek, "Wikiqa: A challenge dataset for open-domain question answering," in Proceedings of the 2015 conference on empirical methods in natural language processing, 2015, pp. 2013–2018.

[47] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," Neural Comput Appl, vol. 27, pp. 1053–1073, 2016.

[48] Y. Shen et al., "Knowledge-aware attentive neural network for ranking question answer pairs," in The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018, pp. 901–904.

[49] S. Kamath, B. Grau, and Y. Ma, "Predicting and integrating expected answer types into a simple recurrent neural network model for answer sentence selection," Computación y Sistemas, vol. 23, no. 3, pp. 665–673, 2019.

[50] S. V. Moravvej, M. J. M. Kahaki, M. S. Sartakhti, and A. Mirzaei, "A method based on attention mechanism using bidirectional long-short

term memory (BLSTM) for question answering," in 2021 29th Iranian Conference on Electrical Engineering (ICEE), IEEE, 2021, pp. 460–464.

[51] Y. Matsubara, T. Vu, and A. Moschitti, "Reranking for efficient transformer-based answer selection," in Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, 2020, pp. 1577–1580.

[52] S. Kim, I. Kang, and N. Kwak, "Semantic sentence matching with densely-connected recurrent and co-attentive information," in Proceedings of the AAAI conference on artificial intelligence, 2019, pp. 6586–6593.

[53] Y. Song, Q. V. Hu, and L. He, "P-CNN: Enhancing text matching with positional convolutional neural network," Knowl Based Syst, vol. 169, pp. 67–79, 2019.

[54] G. Bao, Y. Wei, X. Sun, and H. Zhang, "Double attention recurrent convolution neural network for answer selection," R Soc Open Sci, vol. 7, no. 5, p. 191517, 2020.

[55] Q. Wang, W. Wu, Y. Qi, and Z. Xin, "Combination of active learning and self-paced learning for deep answer selection with bayesian neural network," in ECAI 2020, IOS Press, 2020, pp. 1587–1594.

[56] W. Huang, Q. Qu, and M. Yang, "Interactive knowledge-enhanced attention network for answer selection," Neural Comput Appl, vol. 32, pp. 11343–11359, 2020.

[57] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," Journal of global optimization, vol. 11, pp. 341–359, 1997.

[58] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," Advances in engineering software, vol. 69, pp. 46–61, 2014.

[59] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in Nature inspired cooperative strategies for optimization (NICSO 2010), Springer, 2010, pp. 65–74.

[60] D. Bairathi and D. Gopalani, "Salp swarm algorithm (SSA) for training feed-forward neural networks," in Soft Computing for Problem Solving: SocProS 2017, Volume 1, Springer, 2019, pp. 521–534.

[61] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in 2009 World congress on nature & biologically inspired computing (NaBIC), Ieee, 2009, pp. 210–214.

[62] S. Mirjalili and A. Lewis, "The whale optimization algorithm," Advances in engineering software, vol. 95, pp. 51–67, 2016.

[63] J. Cherian, "Determining the amount of earthquake displacement using differential synthetic aperture radar interferometry (D-InSAR) and satellite images of Sentinel-1 A: A case study of Sarpol-e Zahab city," Advances in Engineering and Intelligence Systems, vol. 1, no. 01, 2022.

[64] G. Hackeling, Mastering Machine Learning with scikit-learn. Packt Publishing Ltd, 2017.

[65] S. Sonkar, A. E. Waters, and R. G. Baraniuk, "Attention word embedding," arXiv preprint arXiv:2006.00988, 2020.

[66] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[67] B. Athiwaratkun, A. G. Wilson, and A. Anandkumar, "Probabilistic fasttext for multi-sense word embeddings," arXiv preprint arXiv:1806.02901, 2018.