

Article

Educational QA System-Oriented Answer Selection Model Based on Focus Fusion of Multi-Perspective Word Matching

Xiaoli Hu, Junfei He, Zhaoyu Shou, Ziming Liu and Huibing Zhang * 

Guangxi Key Lab of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China

* Correspondence: zhanghuibing@guet.edu.cn

Abstract

Question-answering systems have become an important tool for learning and knowledge acquisition. However, current answer selection models often rely on representing features using whole sentences, which leads to neglecting individual words and losing important information. To address this challenge, the paper proposes a novel answer selection model based on focus fusion of multi-perspective word matching. First, according to the different combination relationships between sentences, focus distribution in terms of words is obtained from the matching perspectives of serial, parallel, and transfer. Then, the sentence's key position information is inferred from its focus distribution. Finally, a method of aligning key information points is designed to fuse the focus distribution for each perspective, which obtains match scores for each candidate answer to the question. Experimental results show that the proposed model significantly outperforms the Transformer encoder fine-tuned model based on contextual embedding, achieving a 4.07% and 5.51% increase in MAP and a 1.63% and 4.86% increase in MRR, respectively.

Keywords: question answering (QA) system; answer selection; multi-perspective; matching focus; key information



Academic Editor: Paolo Bellavista

Received: 25 June 2025

Revised: 26 August 2025

Accepted: 8 September 2025

Published: 19 September 2025

Citation: Hu, X.; He, J.; Shou, Z.; Liu, Z.; Zhang, H. Educational QA System-Oriented Answer Selection Model Based on Focus Fusion of Multi-Perspective Word Matching. *Computers* **2025**, *14*, 399. <https://doi.org/10.3390/computers14090399>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Community question answering (CQA) platforms are very popular for learning due to their flexibility and convenience. However, low-quality questions and irrelevant answers often plague these platforms, leading to a frustrating experience for users, who spend a significant amount of time retrieving useful information. Answer selection techniques have been developed in CQA platforms to filter out low-quality or low-matching answers. For example, Stack Overflow employs a recently optimized answer selection algorithm that rapidly recommends high-quality answers to users [1]. However, QA requires recognition of intention and logical reasoning capabilities, which makes it more difficult than other natural language processing tasks such as text classification, machine translation, and sequence labeling [2].

The current educational model is transforming towards openness and diversity, which promotes the continuous iterative updates of CQA. The open and diverse teaching characteristics necessitate that answer selection technology pays more attention to interactive effects and the tracing of complex causal relationships, which further tests the model's intent recognition and reasoning abilities [3]. One of the most effective ways to enhance these abilities is through interaction, which can build communication between sentences, ensuring the consistency of information and improving the model's representation ability of the text [4]. However, the advantage of interaction is limited to expanding the scope of

information from a single sentence to multiple sentences, and cannot endow the model with general semantic understanding ability. To address this issue, pre-trained models such as ELMo [5], BERT [6], GPT [7] and the general purpose large language models they derive have emerged, which store a considerable amount of general language information through parameter storage before downstream tasks are given. Therefore, a question answering system with a pre-training model has a stronger sentence comprehension ability. However, most QA statements, especially in open-domain QA [8], contain multiple and correlated features that cannot be addressed by interaction techniques and pre-training methods alone. In recent years, many scholars have proposed multi-process and multi-level analysis models for answer selection tasks, inspired by the human-like thinking of reasoning and analysis that involves multiple points, aspects, and processes. These models incorporate additional processes for sentence information extraction and feature recognition, naturally enhancing the reasoning abilities of the models. However, existing answer selection models typically rely on sentence-level features, which may obscure the information of key vocabulary in the sentence. For example, in the question “When and where did Lucy and Bob meet in London and for how long did they chat?”, there are three key points: “when”, “where”, and “how long”. However, if the analysis is conducted based on the entire sentence, the model may overlook one or more of these key points.

This paper proposes an answer selection model called the Fusing Multi-perspective Word Matching Focus (FMWMF) model, which is based on sentence words. Drawing inspiration from the current multi-process and multi-level analysis approach, the FMWMF model calculates the focus distribution results of the question and answer from three matching perspectives, serial, parallel, and transfer between QA sentences, and then calculates the matching degree between questions and answers from the information focus distribution of vocabulary. The model not only complements the idea of answer selection algorithm research from vocabulary, but also makes it easier for the model to focus on the key information inside the sentence and complete the tracing of complex causal relationships. Next, we will introduce some related work in Section 2. Section 3 describes the model structure in detail, Section 4 verifies the model effect, and Section 5 summarizes the paper.

2. Related Work

QA systems have gained popularity across various fields, with educational QA systems being a prime example that has attracted extensive research attention. For instance, in study [9], a QA system was designed for high school education by integrating knowledge graphs, intelligent QA, and big data technology. This system not only provides timely and accurate answers but also offers feedback on students’ learning progress. Meanwhile, study [10] proposes an intelligent QA model based on user search behavior, which enhances the model’s ability to extract keywords by utilizing the regularities between user cognition and library elements, thereby improving the quality of the generated answers. In addition, some scholars have applied QA models to automatic grading systems for subjective questions [11]. The foundation for implementing these QA systems is answer selection, which is crucial. For example, in study [12], a deep learning-based intelligent QA system was proposed for railway technical specifications, which accurately understands user intent in a specific domain and improves the success rate of semantic matching. Meanwhile, study [13] points out that, while the answer selection model based on deep neural networks is better than most machine learning methods, machine learning methods have strong interpretability [14–16], which can assist deep learning models in improving the rationality of answers. Early deep learning models lacked interactivity. For instance, two studies [17,18] proposed answer selection models based on LSTM and CNN, respectively, both of which focused on

representing sentence features while ignoring the role of interaction in sentence matching. However, with the introduction of compare-aggregate frameworks into sentence matching models [4], many answer selection models have started to utilize different interaction mechanisms to enhance the model's ability to represent sentences. For instance, study [19] proposes a bidirectional matching model for questions and answers, enabling the model to identify sentence matching levels from different sentences. Additionally, study [20] utilizes self-attention mechanisms and the RoBERTa model to build a health education system, achieving impressive results in non-descriptive question answer generation. Of course, the idea of interaction also directly affects the training methods of today's large models [21] and API calls [22]. Apart from interaction, adding pre-training structures is an effective approach to enhancing the model's ability to represent sentences. For example, study [23] uses sample replacement instead of masking to train language models, resulting in similar results to BERT with less than a quarter of the computational cost. Meanwhile, study [24] utilizes contextual word embeddings and a Transformer encoder (CETE) to model sentence similarity in answer selection. The authors proposed an answer selection model based on fine-tuning BERT and achieved the best answer selection performance on RoBERTa. At present, the most popular approach is to use general purpose large language models and set appropriate prompts to complete the automatic judgment and output of answers [25].

Answer selection models currently employ multi-process and multi-level strategies to mimic human reasoning processes from shallow to deep levels. For instance, study [26] proposed a staged reading comprehension model that involves rough reading to obtain key information points, followed by reinforced reading to verify the answer and make the final prediction. In most cases, a question has more than one key information point, and the answer must contain multiple points related to it. To address this, study [8] used a multi-hop network that iteratively scanned each possible information point in the sentence using attention mechanisms and summed up the matching results to obtain the final QA matching score. Additionally, study [27] proposed a multi-segment interaction matching model that interacts with different segment combinations, capturing more enriched semantics than previous models that interact with isolated words. Furthermore, study [28] proposed a sentence matching model with multi-turn reasoning that focuses on matching features in each turn and uses a memory component to connect the reasoning results of multiple rounds, providing the ability to reason through complex problems at multiple levels. At present, the most modified multi-round reasoning method is COT based on large language models, which completes a very challenging task by role-playing multiple large models [29]. Alternatively, a single MoE-structured model trained with a tailored reinforcement-learning regimen can be employed to obtain exceptional long-horizon reasoning capabilities [30].

3. Model Architecture

Figure 1 illustrates the overarching framework of FMWMF. In this framework, the input question $Q = \{w_1^q, w_2^q, w_3^q, \dots, w_n^q\}$ is assumed to have a length of n , while the input answer $A = \{w_1^a, w_2^a, w_3^a, \dots, w_m^a\}$ is assumed to have a length of m . Here, w_n^q and w_m^a represent the n -th word in the question and the m -th word in the answer, respectively. The first step is to input the (Q,A) pair into three different models: the serial matching model, parallel matching model, and transformational matching model. These three models map sentence-to-sentence coherence, relevance, and logical information into the vocabulary in the sentence, and form an information focus. Subsequently, we extract key information points and their positions in the sentence based on the focus distribution results. Finally, we fuse the word focus matching results from different matching perspectives to obtain the matching score of the input question–answer pair. Based on this score, we can compare and sort the potential answers, ultimately arriving at the best answer. The

detailed implementation process of the entire model can be found in Algorithm 1 and the corresponding model design schematics diagram, as shown in Figure 2.

Algorithm 1: Pseudo-code for implementing the FMWMF

Input: train set $T = \{(q, a, lab)\}$, test set $V = \{(q, a, lab)\}$, hyperparameters of each matching model H_s, H_p, H_t , and the epoch e .

- 1: $\theta_s, \theta_p, \theta_t \leftarrow$ Initialize model parameters
- 2: #Step 1: train matching models respectively
- 3: for epoch $\in e$ do
- 4: for $\{(q, a, lab)\} \in T_{\text{mini_batch}}$ do
- 5: $\{o_s\}_{\text{mini_batch}} = f_{\theta_s}(\{q, a\}_{\text{mini_batch}}), l = \text{loss}_s(\{o_s\}_{\text{mini_batch}}, \{lab\}_{\text{mini_batch}});$
- 6: $\{o_p\}_{\text{mini_batch}} = f_{\theta_p}(\{q, a\}_{\text{mini_batch}}), l = \text{loss}_p(\{o_p\}_{\text{mini_batch}}, \{lab\}_{\text{mini_batch}});$
- 7: $\{o_t\}_{\text{mini_batch}} = f_{\theta_t}(\{q, a\}_{\text{mini_batch}}), l = \text{loss}_t(\{o_t\}_{\text{mini_batch}}, \{lab\}_{\text{mini_batch}});$
- 8: Update $\theta_s, \theta_p, \theta_t$ by backpropagation
- 9: end for
- 10: $\{(q, a, lab)\} \in V;$
- 11: $\{o_s\}_{\text{test}} = f_{\theta_s}(\{q, a\}_{\text{test}}), acc_s^* = \sum \text{comp}(\{o_s\}_{\text{test}}, \{lab\}_{\text{test}});$
- 12: $\{o_p\}_{\text{test}} = f_{\theta_p}(\{q, a\}_{\text{test}}), acc_p^* = \sum \text{comp}(\{o_p\}_{\text{test}}, \{lab\}_{\text{test}});$
- 13: $\{o_t\}_{\text{test}} = f_{\theta_t}(\{q, a\}_{\text{test}}), acc_t^* = \sum \text{comp}(\{o_t\}_{\text{test}}, \{lab\}_{\text{test}});$
- 14: $acc_s, \theta_s = \max_{\theta_s}(acc, acc_s^*);$
- 15: $acc_p, \theta_p = \max_{\theta_p}(acc, acc_p^*);$
- 16: $acc_t, \theta_t = \max_{\theta_t}(acc, acc_t^*);$
- 17: end for
- 18: Save $\theta_s, \theta_p, \theta_t;$
- 19: #Step 2: Utilize each matching model to extract the focus distribution of words in a sentence, and output the question–answer matching score.
- 20: $dis_s = d_{\theta_s}(q, a), dis_p = d_{\theta_p}(q, a), dis_t = d_{\theta_t}(q, a);$
- 21: $x_s, x_p, x_t \in R^k = \text{topK}(dis_s, dis_p, dis_t);$
- 22: $s = \frac{\sum_i \sum_{m1, m2 \in \{s, p, t\}} x_{m1}[i] - x_{m2}[i]}{3K}$
- 23: Output s

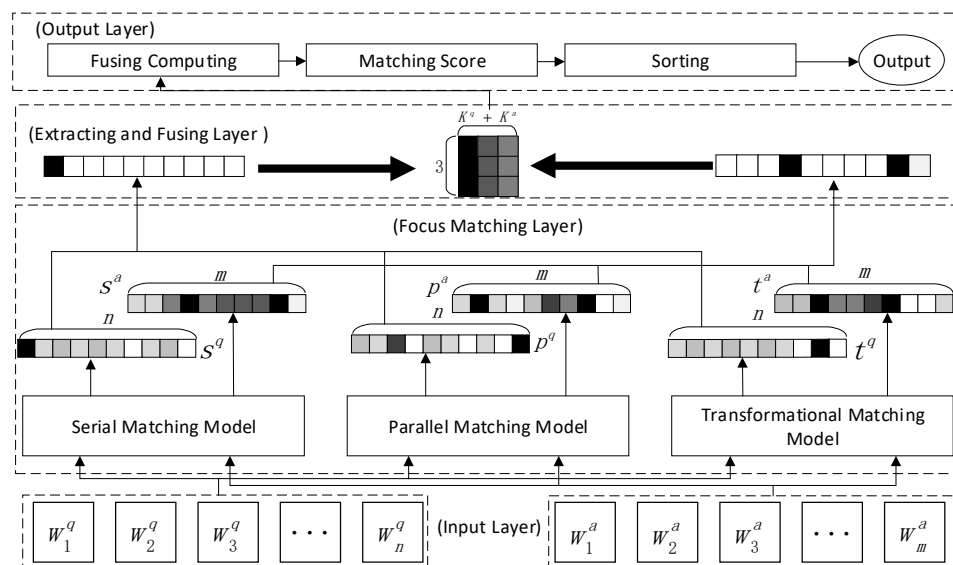


Figure 1. Overall structure of FMWMF.

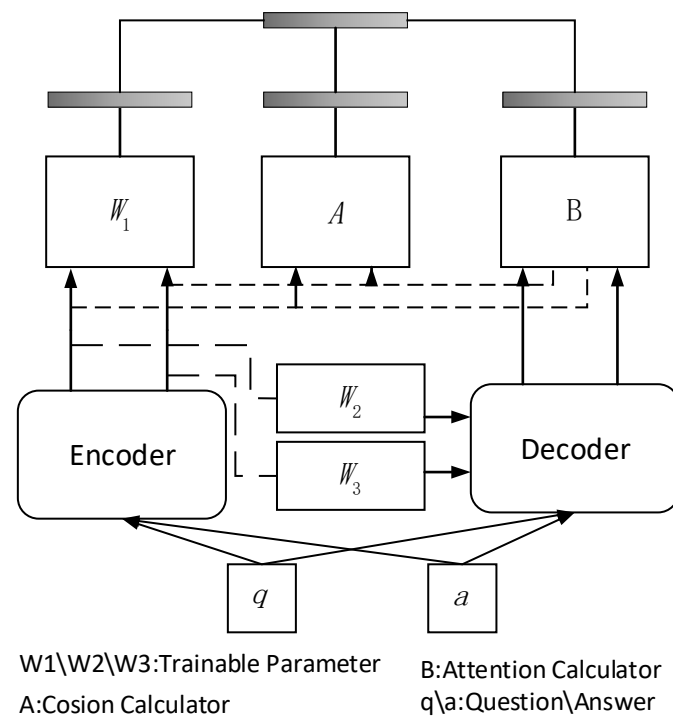


Figure 2. Model design schematics diagram.

3.1. Serial Matching Structure

Serial matching is a technique used in QA that involves combining a question and its corresponding answer into a single word sequence. This concatenated sequence is then fed into a classifier model to determine whether the two sentences form a valid QA pair. Figure 3 illustrates the architecture of the serial matching model used for QA.

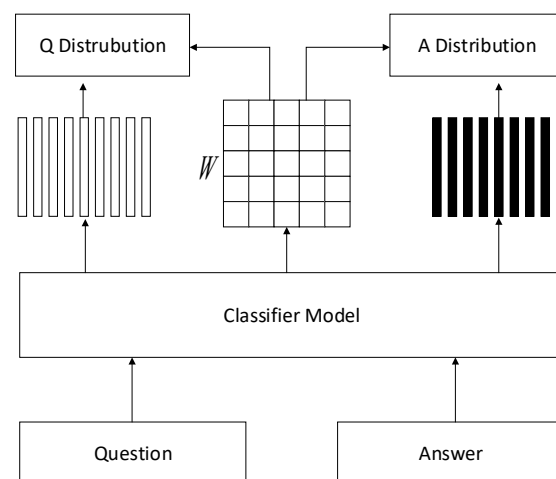


Figure 3. Structure of the serial matching model.

In this approach, a classifier model based on BERT is utilized. To begin, the input consists of a question $Q = \{w_1^q, w_2^q, \dots, w_n^q\}$ with a length of n and an answer $A = \{w_1^a, w_2^a, \dots, w_m^a\}$ with a length of m . The n -th word in the question sentence and the m -th word in the answer sentence are represented as w_n^q , w_m^a and, respectively. The question and answer are then concatenated using [CLS] and [SEP] tokens to create a token sequence, which is then passed through the BERT model to generate a feature vector

$c \in R^{1 \times d}$ for the classifier. Additionally, dynamic word vectors $q^c \in R^{n \times d}$, $a^c \in R^{m \times d}$ are obtained for the question and answer sentences, as demonstrated in Equation (1):

$$c, q^c, a^c = bert([cls], w_1^q, w_2^q, \dots, w_n^q, [sep], w_1^a, w_2^a, \dots, w_m^a) \quad (1)$$

After obtaining the classification feature vector, the next step is to use a multilayer perceptron (MLP) model to map it to a binary classification label. This is represented by Equation (2):

$$y = c \cdot W_1 \cdot W_2 \quad (2)$$

where $y \in R^{1 \times 2}$ represents the prediction vector, in which the first element of the vector represents the probability that there is no correlation between sentences, and the second element represents the probability that there is a correlation. $W_1 \in R^{d \times k}$ and $W_2 \in R^{k \times 2}$ are parameter matrices that need to be learned. The loss function for training the serial matching model is shown in Equation (3):

$$l = -((1 - y_{label}) \log y_1 + y_{label} \log y_2) \quad (3)$$

where y_1, y_2 respectively represent the first and second elements of the vector, and $y_{label} \in \{0, 1\}$ represents the true label of whether there is a relationship between the question–answer pairs. W_1, W_2 records the correlation information between the question–answer sentences. Then, we can calculate the focus of words in a sentence with W_1, W_2 and q^c, a^c , mapping each word to a result related to the classification, as shown in Equations (4) to (5):

$$y^q = q^c \cdot W_1 \cdot W_2 \quad (4)$$

$$y^a = a^c \cdot W_1 \cdot W_2 \quad (5)$$

where $y^q \in R^{n \times 2}$, $y^a \in R^{m \times 2}$ and because the matching focus is on the contribution of each word to the relationship, the final matching focus distribution between the question and answer is shown in Equations (6) to (7):

$$s^q = y^q[:, 1] \quad (6)$$

$$s^a = y^a[:, 1] \quad (7)$$

3.2. Parallel Matching Structure

Parallel matching is a technique that uses word embedding models to obtain embeddings for both the question and answer sentences, as well as for individual words. This approach allows us to calculate the cosine similarity between the sentence embedding of the question and each vector of words in the answer, resulting in the focus matching distribution of the answer. Similarly, we can derive the distribution of the word focus of the question. Figure 4 illustrates the structure of the parallel matching approach.

The embedding model is based on SBERT, which is typically trained using two structures: either a Siamese network or triplet network [31]. Here, we directly utilize pre-trained open-source models and their corresponding parameters available on the internet. Regarding question $Q = \{w_1^q, w_2^q, \dots, w_n^q\}$ and answer $A = \{w_1^a, w_2^a, \dots, w_m^a\}$, inputting them into the SBERT will yield the sentence embeddings for both the question and answer, as well as the word embeddings for each word in the sentence, as shown in Equations (8) and (9):

$$e^q = sbert(w_1^q, w_2^q, \dots, w_n^q) \quad (8)$$

$$e^a = sbert(w_1^a, w_2^a, \dots, w_m^a) \quad (9)$$

where $e^q \in R^{1 \times d}$, $e^a \in R^{1 \times d}$ are the embedding results of sentences, and d represents the dimension of the sentence embedding vector. After inputting the question and answer into the BERT model, the dynamic word embeddings of each word can be obtained, as shown in Equations (10) and (11):

$$o^q = \text{bert}(w_1^q, w_2^q, \dots, w_n^q) \quad (10)$$

$$o^a = \text{bert}(w_1^a, w_2^a, \dots, w_m^a) \quad (11)$$

where $o^q \in R^{d \times n}$ represents the dynamic word embedding vector of each word in the question sentence, and $o^a \in R^{d \times m}$ represents the dynamic word embedding vector of each word in the answer sentence. After obtaining e^q , e^a , o^q , o^a , the word matching focus distribution under parallel structure can be obtained through Equations (12) and (13):

$$p^q = (e^a \cdot o^q) \odot |o^q| \quad (12)$$

$$p^a = (e^q \cdot o^a) \odot |o^a| \quad (13)$$

where p^q and p^a respectively represent the question word focus distribution result and the answer word focus distribution result; $|o^q| \in R^{1 \times n}$, and its n -th element is $|o^q|_n = \frac{1}{|o^q|_{:n}||e^a|}$, and the values of each element in $|o^a|$ can be obtained in the same way; $\odot$ represents element-wise multiplication.

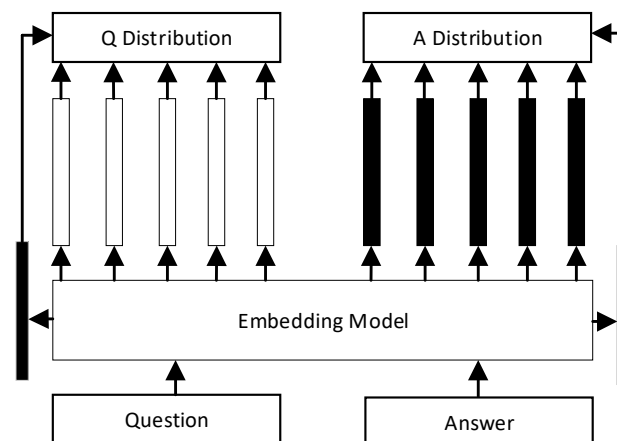


Figure 4. Structure of the parallel matching model.

3.3. Transformational Matching Structure

Transformational matching is a technique that achieves question–answer matching through translation between questions and answers. The general process of this matching is shown in Figure 5.

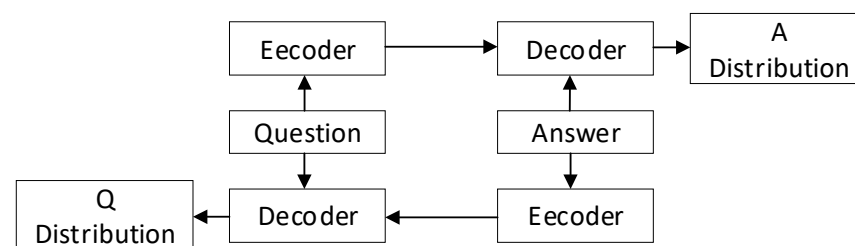


Figure 5. Structure of the transformational matching model.

The matching structure shown in Figure 4 includes two seq2seq models, each of which contains an encoder and a decoder. Using the above structure, the computation of the question focus distribution result is a process of answer-to-question transformation. The

first step is to encode the answer $A = \{w_1^a, w_2^a, \dots, w_m^a\}$, and the encoding result is shown in Equation (14):

$$o_a^{en} = \text{encoder}(w_1^a, w_2^a, \dots, w_m^a) \quad (14)$$

where o_a^{en} represents the encoding matrix of the answer, h_a^{en} and denotes the feature vector of the answer. The encoding matrix o_a^{en} output by the encoder and the feature vector h_a^{en} are transformed into decoding states through two fully connected layers, as shown in Equation (15):

$$o_a^{de} = w_o^{de} \cdot o_a^{en} \quad (15)$$

where o_a^{de}, h_a^{de} are the encoding matrix and feature matrix vector of the answer in the decoding state. $W_o^{de} \in R^{h_d \times h_e}$ and $W_h^{de} \in R^{h_d \times h_e}$ are the parameter matrices that need to be learned; h_d, h_e are the sizes of the decoder and encoder hidden layers. After encoding the input answer text, h_a^{de} it is input to the decoder layer, and the input question $Q = \{w_1^q, w_2^q, \dots, w_n^q\}$ is encoded using the GRU component in the decoder, resulting in the decoding matrix and feature vector o_q^{de}, h_q^{de} for the question, as shown in Equation (16):

$$o_q^{de} = \text{decoder}(w_1^q, w_2^q, \dots, w_n^q, o_a^{de}) \quad (16)$$

Afterwards, the outputs o_q^{de}, o_a^{de} from the encoder and decoder are used to calculate the attention level of each word in the question on the answer, as shown in Equation (17):

$$att = o_q^{de} \cdot o_a^{de} \quad (17)$$

The encoding matrix o_a^{en} of the answer in the encoding state is transformed into the encoding matrix o_q^{en} of the question in the encoding state after passing through the attention layer, as shown in Equation (18):

$$o_q^{en} = att \cdot o_a^{en} \quad (18)$$

Then, using Equation (19), we can obtain the focus distribution of the question.

$$t^q = o_a^{en} \cdot o_q^{en} \quad (19)$$

where t^q represents the word matching focus distribution of the question under the transformational matching structure. Finally, to train the parameters W_o^{de}, W_h^{de} , the loss value needs to be calculated using the decoder's output and the true question. The state matrices of the question encoding state and the decoding state are connected to the encoding matrix o_q^{en}, o_q^{de} , and after passing through two fully connected layers, the decoding output w_{out} of the answer is obtained, as shown in Equation (20):

$$w_{out} = W^{out} \cdot (W^{de} \cdot (o_q^{en}; o_a^{de})) \quad (20)$$

where $w_{out} \in R^{1 \times |V|}$ ($|V|$ represents the size of the vocabulary) $W^{out} \in R^{|V| \times |h'|}$ and $W^{de} \in R^{|V| \times |h'|}$ are both parameter matrices that need to be learned, and $|h'|$ is the size of the hidden layer output by the first fully connected layer. Finally, the training sentence-level word pair matching loss function L is shown in Equation (21):

$$L = \frac{-\sum_{i=0}^n \log \text{softmax}(w_{out}^i)[p_i]}{n} \quad (21)$$

where w_{out}^i is the vector of predicted values of all words in the dictionary for the i -th word in decoding, and p_i is the position of the i -th word in the dictionary in the ground truth answer.

Similarly, for the semantic focus distribution of the answer, a seq2seq model is used to convert the question into an answer and obtain the word matching focus distribution t^a during the conversion process.

3.4. Information Extraction and Fusion

After obtaining the results of word matching focal distributions through various matching models, the focus distribution results for the question and answer are $s^q, p^q, t^q, s^a, p^a, t^a$, respectively. Then, the positions of key information points in the sentence are extracted based on the focus distribution results, as shown in Equation (22):

$$x = \text{topK}(f^x) \quad (22)$$

where $\text{topK}(\cdot)$ represents the top K numbers extracted from the vector and outputs the positions of each element $x \in R^K f^x \in \{s^q, p^q, t^q, s^a, p^a, t^a\}$, as represents the resulting focus distribution obtained. Different models have different matching characteristics. The serial matching model tends to look for vocabularies that indicate whether there is a question–answer relationship in the question and answer, the parallel matching model pays more attention to whether there are similar expressions in the question and answer, and the transformational matching model is better at finding vocabulary that indicates a logical connection between the question–answer sentences. However, regardless of which matching model is used to extract key information from the question–answer sentences, the positions of the information should be very close to each other to reflect the stability of language expression. So, based on Equation (22), we can obtain the position distribution of key information points in the question under the three matching modes as x_s^q, x_p^q, x_t^q and x_s^a, x_p^a, x_t^a , and then calculate the distance between the corresponding positions of key information points in the question and answer, respectively, as shown in Equations (23) to (24):

$$d^q = \frac{\sum_{i=1}^K |x_s^q[i] - x_p^q[i]| + |x_s^q[i] - x_t^q[i]| + |x_p^q[i] - x_t^q[i]|}{3 \times K} \quad (23)$$

$$d^a = \frac{\sum_{i=1}^K |x_s^a[i] - x_p^a[i]| + |x_s^a[i] - x_t^a[i]| + |x_p^a[i] - x_t^a[i]|}{3 \times K} \quad (24)$$

Finally, using $d = d^q + d^a$ as the deviation between the two sentences indicating the existence of a question–answer relationship, the smaller the value of d , the higher the possibility of a question–answer relationship between the two sentences.

4. Experiments and Results Analysis

4.1. Experimental Dataset and Evaluation Metrics

Experimental Dataset We selected two typical QA datasets for testing: TREC-QA [32] and Wiki-QA [33]. The basic information about these two datasets is shown in Table 1, with all values obtained after removing entirely negative answers.

Table 1. Information on the dataset.

Dataset	TREC-QA	Wiki-QA
Train Set	1162	873
Validate Set	65	126
Test Set	68	243

Evaluation Metrics The evaluation metrics for the experiment are mean average precision (MAP) and mean reciprocal rank (MRR)

MAP This refers to the mean average precision (MAP) for all the questions. The expression for the average precision for each question is shown in Equation (25):

$$P_{ave}(q) = \frac{\sum_{k=1}^n p(k) \times r(k)}{m} \quad (25)$$

where $p(k)$ is calculated using Equation (26):

$$p(k) = \frac{\sum_{k=1}^i r(k)}{\sum_{k=1}^j k} \begin{cases} i = 1, 2, 3, \dots, m \\ j = 1, 2, 3, \dots, n \end{cases} \quad (26)$$

where q represents a single question, n represents all the candidate answers for a single question, and m represents the number of correct answers. The value $r(k)$ can be either 0 or 1. When the k -th candidate's answer is the correct answer, $r(k) = 1$; otherwise, $r(k) = 0$. The final expression for MAP is shown in Equation (27):

$$MAP = \frac{\sum_{q=1}^{|Q|} P_{ave}(q)}{|Q|} \quad (27)$$

When all the correct answers are ranked before any incorrect answers, the MAP approaches 1. When all the correct answers for a question are ranked after all the incorrect answers, the MAP approaches 0.

MRR This refers to the average value of the reciprocal number of the position where the first correct answer appears in the list of candidate answers predicted by the question. The calculation method is shown in Equation (28):

$$MRR = \frac{\sum_{q=1}^{|Q|} \frac{1}{rank_q}}{|Q|} \quad (28)$$

where $rank_q$ represents the rank of the first correct answer for the question q among all candidate answers. The MRR value also ranges from 0 to 1 and measures the position of the first correct answer in the list of candidate answers.

Models: We compare four classical answer selection algorithms, including PQSG, a classical probability-based statistical learning method; Finetune-RoBERTa, a fine-tuned model based on RoBERTa; MANS (Multihop Attention Networks), a Multi hop inference model based on attention mechanism; and HR (Hierarchical Ranking), in which the relationship between Q&A sentence pairs is divided into sentences, sentence pairs, and sentence lists, and then the relationship between Q&A sentence is matched hierarchically.

4.2. Experimental Environment

The experiments were performed on a system equipped with an Intel Core i5-9400@2.90GHz six-core CPU, 24 GB of RAM, and an Nvidia GeForce GTX 1660 Ti GPU with 6 GB of memory. For the serial matching, we employed the pre-trained bert-base-uncased

version of BERT, which had two fully connected layers with parameters of (768, 100) and (100, 2), respectively. In the case of parallel matching, we utilized SBert from the all-distilroberta-v1 version available on the Hugging Face website. For the Encoder-Decoder model in transformational matching, we used the Adam optimizer with a learning rate of 0.001 to update the parameters. We trained the word embedding representations based on the question–answer pairs using a batch size of 128 and the sentence-level matching module using a batch size of 64. When processing data with the recurrent neural network model, we set the size of its hidden layer to 128 and kept all other parameters at their default values.

4.3. Comparison of Matching Focus Distribution Results Under Different Question–Answer Matching Models

To verify that the focus of key information should yield similar distribution results under different matching structures, we selected two representative questions from the two datasets. We then identified two answer sentences with similar lengths based on the questions, one of which provided a positive answer that was relevant to the question, while the other presented a negative answer that was unrelated to the question. Table 2 provides detailed descriptions of the sentences used in our study. By comparing the distribution results obtained using different matching structures, we can assess the effectiveness of each method in capturing the key information of the answer relevant to the question.

Looking at Figure 5, it is evident that, when the question is matched with the answer, two key information points are present in the question: one near “How much” and the other near “worth”. This implies that the extracted key points from the question revolve around the price or value of something, and the matching trends across the three models are largely similar at these crucial points. However, when the question is paired with a negative answer to form a sentence pair, the distribution of focus in the question varies significantly across different matching modes. The resulting distribution of focus is unable to extract the key information points, thus leading to a large degree of offset in focus positions observed from different models. Figure 6 shows the focus distribution results of answer sentences. When the answer is a positive response to the question, similar trends can be obtained across different matching structures, especially at the key information position. However, the distribution of negative answers is relatively more chaotic. For instance, the trend of focus changes in the serial matching structure displays a random state and is less able to obtain the position of key information points. Similarly, the length of the second question is 8 words, the length of the positive answer is 19 words, and the length of the negative answer is 20 words. The focal distribution results obtained after combining the questions with positive and negative answers under different matching modes are shown in Figures 8 and 9.

Table 2. Examples of question–answer sentence pairs.

Question	Positive Answer	Negative Answer
How much are the Harry Potter movies worth?	The series also originated many types of tie-in merchandise, making the Harry Potter brand worth in excess of USD 15 billion.	The initial major publishers of the books were Bloomsbury in the United Kingdom and Scholastic Press in the United States.
How deep can deep underwater drilling go?	Deepwater drilling is the process of oil and gas exploration and production at depths of more than 500 feet.	It has been economically infeasible for many years, but with rising oil prices, more companies are investing in this area.

Regarding Example 1, the word lengths are as follows: 8 for the question, 19 for the positive answer, and 20 for the negative answer. The results of the word matching focus distribution for the question and the positive/negative answers in different matching structures are illustrated in Figures 6 and 7.

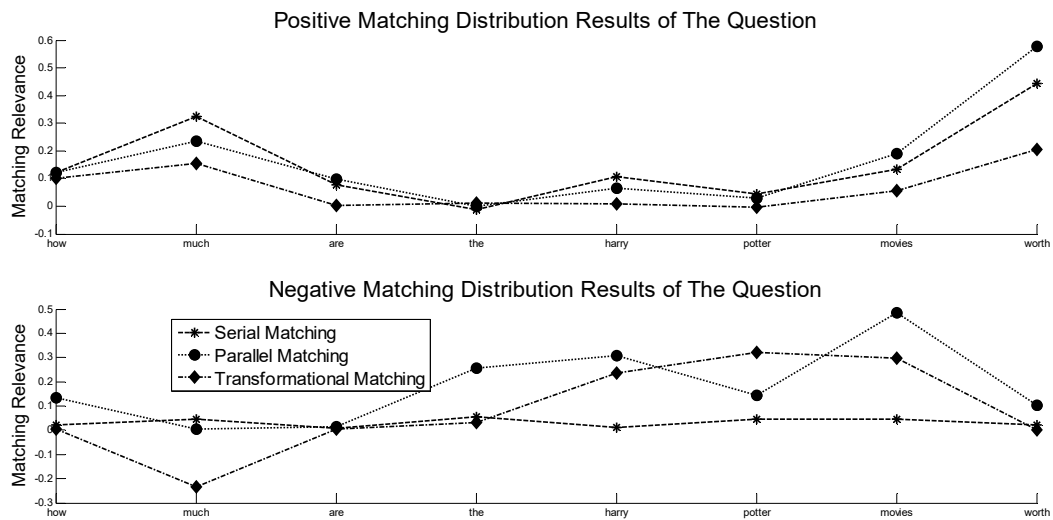


Figure 6. The Q distribution obtained by matching question 1 with positive and negative answers based on word focus.

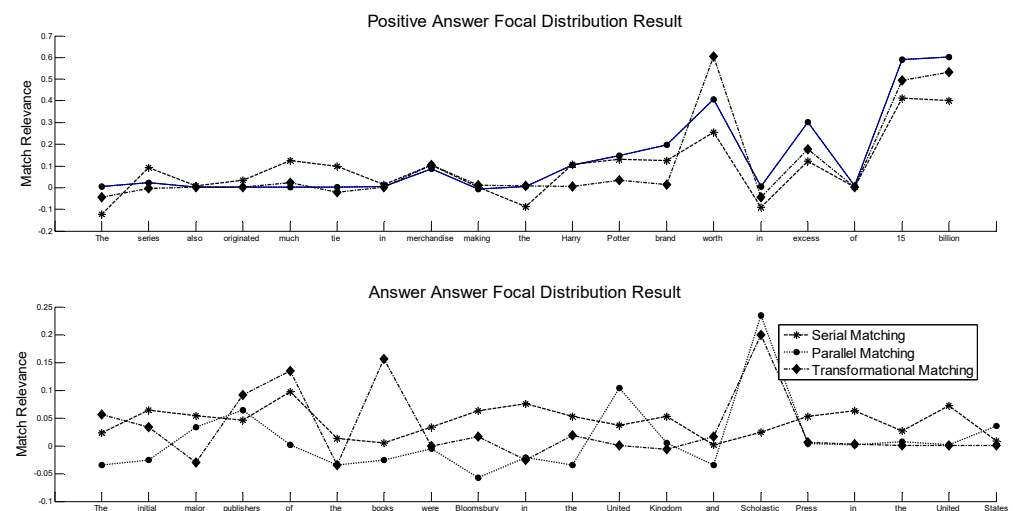


Figure 7. The A distribution obtained by matching question 1 with positive and negative answers based on the word focus.

Figures 8 and 9 suggest a pattern similar to the first example, indicating that the distribution trends and variation rules of the key information positions corresponding to each matching focus under different matching structures are either similar or closely related. To further verify this pattern, we divided the question–answer data in the test set of the two datasets into two categories: positive answers with questions and negative answers with questions. We then calculated the correlation between the distribution of matching focus in different matching perspectives for the two categories of question–answer data, using the Pearson correlation coefficient as the measure. The experimental results are presented in Table 3.

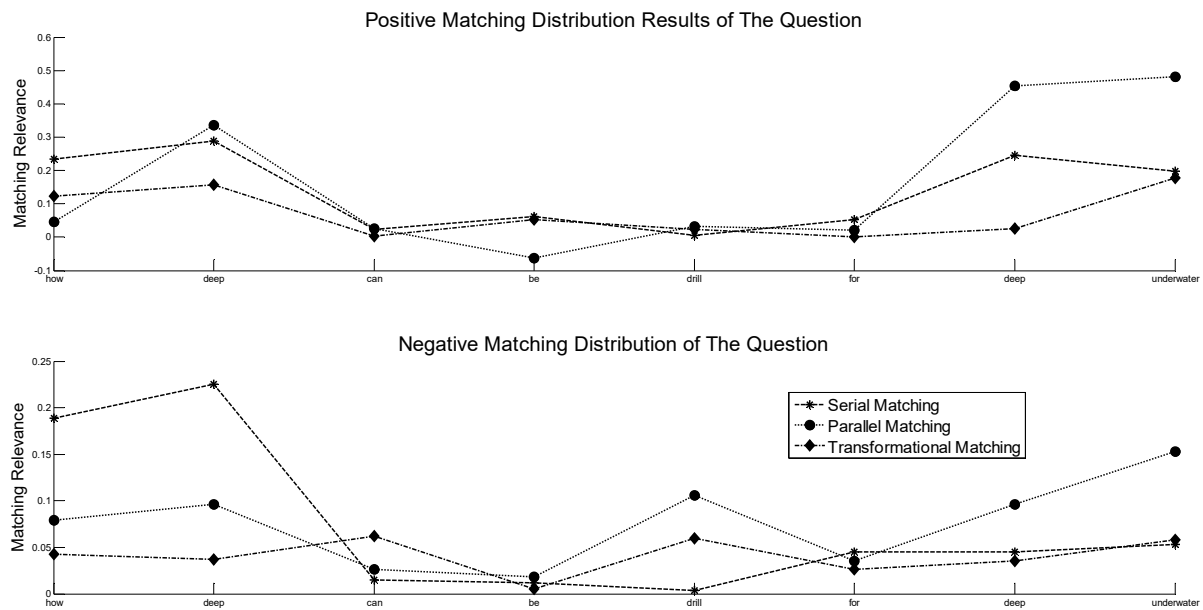


Figure 8. The Q distribution obtained by matching question 2 with positive and negative answers based on word focus.

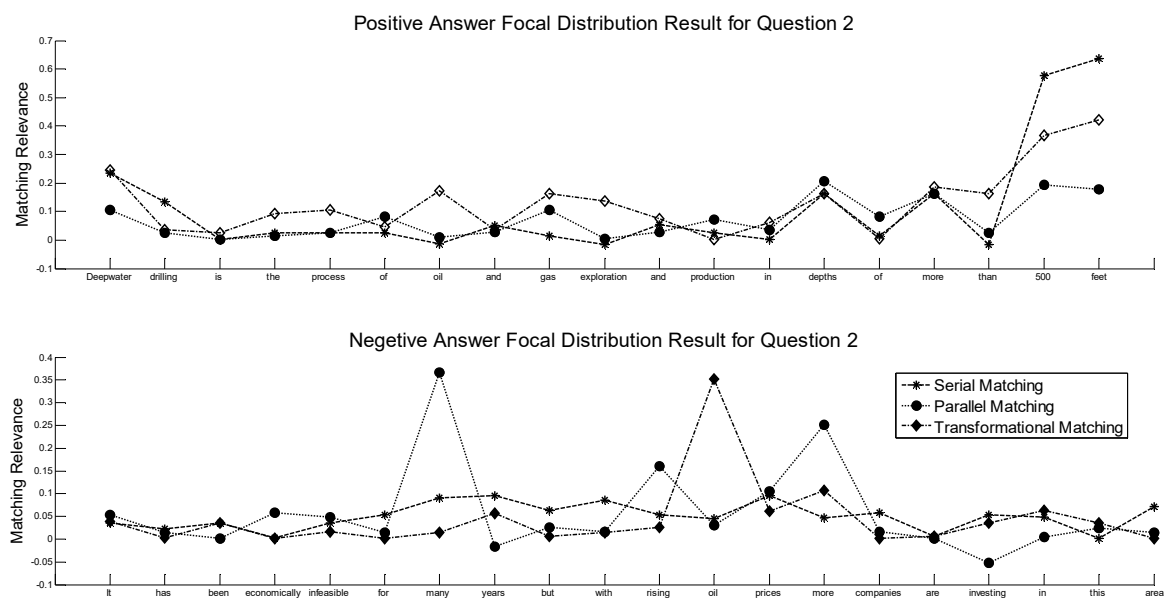


Figure 9. The A distribution obtained by matching question 2 with positive and negative answers based on the word focus.

Table 3. The correlation between the distribution of match focus of positive and negative answers under different matching perspectives.

Dataset	Answer Type	Relevance
TREC-QA	Positive	0.7929
	Negative	−0.1926
Wiki-QA	Positive	0.6322
	Negative	0.2437

Based on the experimental results in Table 3, we can conclude that the distribution of word matching focus for positive answers exhibits a similar trend across different

perspectives, whereas the distribution of word matching focus for negative answers appears to be more random. This finding confirms that the location distribution of key information points for two sentences with question–answer relationships under different perspectives is relatively stable.

4.4. The Impact of the Number of Key Information K on the Effectiveness of Answer Selection

A sentence may contain more than one key information point, and it is not always the case that all words in a sentence are key information points. Hence, it is necessary to assess the ability of questions and answers to include a certain number of information points and the effectiveness of answer selection under different numbers of extracted information points. In this regard, the value K is used to represent the number of extracted key information points in the sentence. Initially, the K value in the answer is fixed to 4, based on previous research findings [34]. Then, the K value in the question is varied to observe the model's effectiveness. Subsequently, the K value in the question is fixed at $K = 2$ (based on the best effectiveness obtained in the previous experiment), and the K value in the answer is changed to observe the model's effectiveness. The experimental results are depicted in Figures 10 and 11.

Based on the relationship between the model's answer selection performance and the change of K value in Figures 9 and 10, we can infer that, for the TREC-QA and Wiki-QA datasets, questions usually have two key information points. In contrast, the key information points in answer choices need to be determined based on the specific category of the question-answering dataset. The optimal performance is generally achieved around $K = 4$, which indicates that answers usually require twice as much information to explain compared to questions, with two key information points in questions and four in answers. Furthermore, we observe that the effectiveness of answer selection shows an upward trend followed by a downward trend as the K value increases. Once the optimal performance is achieved, increasing the K value in the question leads to a monotonic decrease in answer selection performance, while increasing the K value in answer matching leads to a fluctuating decrease in answer selection performance. This also suggests that the distribution of information in answers is more complex from another perspective.

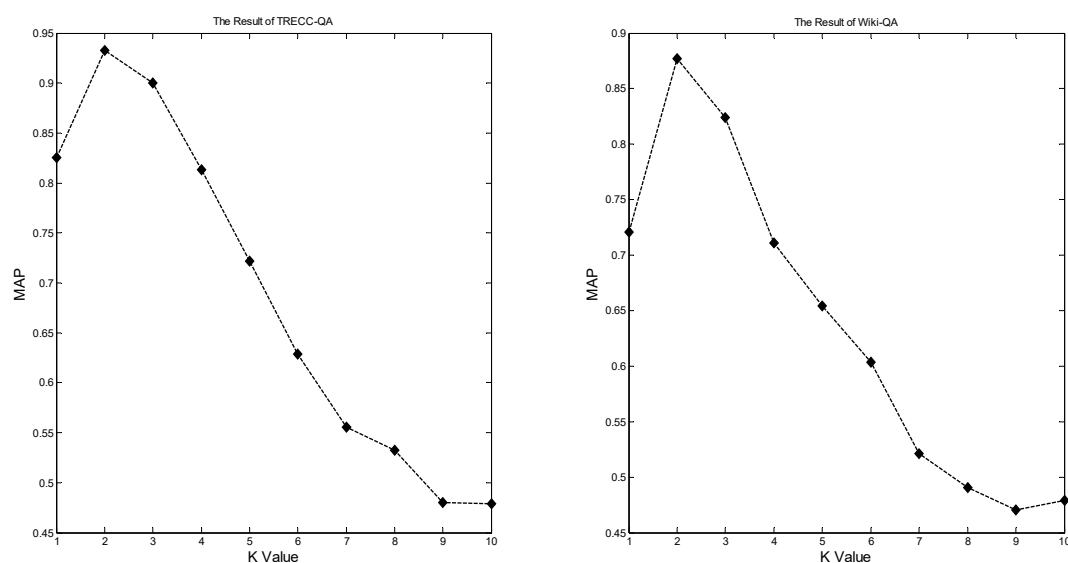


Figure 10. The relationship between different K values in the question and the effectiveness of answer selection.

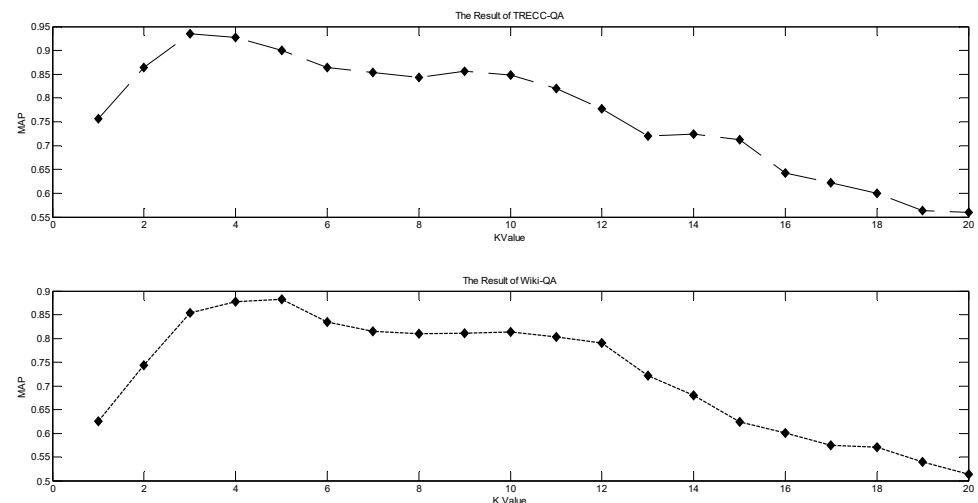


Figure 11. The relationship between different K values in the answer and the effectiveness of answer selection.

4.5. Comparison of the Effect of Different Answer Selection Models

4.5.1. Comparison of the Classical Answer Selection Models

Based on the aforementioned findings, it can be inferred that different matching models have similar patterns in extracting word focus for sentences with question–answer relationships. Hence, any combination of two matching models can be employed for answer selection. To verify the complementary roles of different matching models in extracting sentence information, we combined the three matching models in pairs, denoted as S for serial matching, P for parallel matching, and T for transformational matching mode. For this purpose, we fixed $K = 2$ for questions and $K = 4$ for answers. The answer selection performances of the proposed models on the two question-answering datasets under different combination forms are summarized in Tables 4 and 5, respectively. These tables also include comparison results between our proposed model and some typical answer selection models used in the past.

Table 4. Comparison of experimental results on the TRECQA dataset.

Model	MAP	MRR
Study [8]	0.8130	0.8930
Study [17]	0.7134	0.7913
Study [24]	0.8910	0.9250
Study [35]	0.8420	0.9040
SP	0.8497	0.8738
ST	0.9062	0.9246
TP	0.8805	0.9027
SPT(FMWMF)	0.9273	0.9401

Tables 4 and 5 present the comparison results of FMWMF with other models and different matching perspectives combinations on the TREC-QA and Wiki-QA datasets. Compared to the CETE model, FMWMF has a significant improvement in MAP indicators on both datasets, with an increase of 4.07% on the TREC-QA dataset and 5.51% on the Wiki-QA dataset. The performance improvement on the Wiki-QA dataset is particularly noteworthy, as it covers a wider range of domains and answer formats, indicating that FMWMF can distinguish more subtle differences between sentences in identifying different answers to the same question and reasonably extend and expand based on the question. This allows the model to still make a question–answer relationship judgment, even when

the question and answer have fewer similar and connecting words. For the MRR indicator, FMWMF achieved an improvement of 1.63% on the TREC-QA dataset and 4.86% on the Wiki-QA dataset, demonstrating its enhanced ability to select the best answer. The word-based modeling approach can extract more comprehensive sentence information, and the model also has stronger understanding and reasoning capabilities due to the introduction of the multi-perspective matching mechanism. Moreover, none of the models resulting from the arbitrary combination of the three matching modes can surpass the effect of the simultaneous combination of the three matching models, indicating that the three matching models complement each other in extracting the question–answer relationship between sentences. Looking at the results of the combined experiments, the transformational matching has the strongest ability to extract the question–answer relationship, followed by the serial matching, and finally the parallel matching.

Table 5. Comparison of experimental results on the WikiQA dataset.

Model	MAP	MRR
Study [8]	0.7220	0.7380
Study [24]	0.8290	0.8430
Study [32]	0.6520	0.6652
Study [34]	0.7420	0.7540
SP	0.7934	0.7977
ST	0.8306	0.8361
TP	0.8147	0.8213
SPT(FMWMF)	0.8773	0.8840

4.5.2. Comparison of the Large Language Models

In order to verify the use of the model in the real environment, we selected the latest question and answer dataset in natural language processing, mmlu_pro, which contains 14 question and answer questions in different fields, one of which corresponds to at least 10 answers. Answer selection for such a complex task is difficult for the general model to handle and achieve good results. Here we select Deepseek-R1 and qwen3_MoE, two models with reasoning ability, and set the first 100 questions to be selected for each subject, and the output length of the model as 4096. The following are the results of the control experiment, including the accuracy of answer selection, the time spent, and the optimal K value of the FMWMF model.

It can be seen from Table 6 that the K values selected for Q&A data in different fields are different, which is obviously related to the subject data corresponding to the data, but the K values selected by the Q&A datasets in most fields are a combination of (2 and 4). From the perspective of experimental accuracy, the Qwen2.5–1.5B model with the same parameter level as the FMWMF model performs very poorly in any discipline, indicating that LLMs with small parameter levels are not competent for complex tasks; however, for the FMWMF model, the results of the model in the textual reasoning discipline are equivalent to the results of Deepseek-R1 or Qwen3_MoE, and even surpass the two reasoning LLM models in the Law discipline. Finally, in terms of calculation time, FMWMF spends the least time because FMWMF only processes the input questions and answer options, and does not need the prompt and analysis process of the second subject, so the time spent on each subject is basically the same.

Table 6. Comparison of the large language models.

Subject	DeepSeek-R1 (16*H20)		Qwen_MoE- think (8*H20)		Qwen2.5–1.5B (1*H20)		FMWMF (1*H20)			
	acc	Time	acc	Time	acc	Time	acc	Time	K (Question)	K (Answer)
Biology	0.86	40 min 21 s	0.92	30 min 36 s	0.2193	3 min 50 s	0.67	67 s	2	4
Business	0.8	39 min 07 s	0.83	26 min 48 s	0.25	3 min 07 s	0.73	62 s	2	5
Chemistry	0.76	55 min 00 s	0.86	39 min 54 s	0.28	5 min 27 s	0.27	55 s	4	7
Computer Science	0.79	23 min 09 s	0.81	21 min 56 s	0.11	2 min 34 s	0.76	68 s	2	4
Economics	0.85	35 min 13 s	0.87	25 min 41 s	0.31	3 min 27 s	0.64	77 s	2	4
Engineering	0.64	19 min 18 s	0.71	15 min 00 s	0.26	1 min 45 s	0.68	59 s	2	4
Health	0.69	12 min 04 s	0.69	10 min 30 s	0.19	1 min 11 s	0.58	60 s	2	4
History	0.67	36 min 17 s	0.68	25 min 54 s	0.21	3 min 30 s	0.61	78 s	3	6
Law	0.59	38 min 05 s	0.65	29 min 42 s	0.13	3 min 40 s	0.72	93 s	2	4
Math	0.92	34 min 48 s	0.9	27 min 45 s	0.13	3 min 38 s	0.15	51 s	2	2
Other	0.75	12 min 11 s	0.65	9 min 51 s	0.34	1 min 09 s	0.70	61 s	2	4
Philosophy	0.81	28 min 35 s	0.7	22 min 19 s	0.14	2 min 55 s	0.81	84 s	3	6
Physics	0.87	34 min 19 s	0.9	29 min 23 s	0.24	3 min 35 s	0.22	56 s	2	6
Psychology	0.72	11 min 26 s	0.79	9 min 59 s	0.19	1 min 10 s	0.75	63 s	3	6

5. Conclusions

This paper proposes a novel answer selection model, FMWMF, that utilizes word focus from multiple perspectives. FMWMF can capture more detailed information in the sentence and solve the problem of information omission. This method enables rapid and accurate retrieval of the target answer from a rich list of candidates while simultaneously extracting the key information points embedded in each question–answer pair. Empirically, the identified answers consistently contain multiple of the information points present in the original query. Deploying FMWMF in community-based and educational Q&A platforms would therefore drastically accelerate the curation of high-quality answers and mitigate the inefficiencies caused by information redundancy. Nevertheless, we observe that FMWMF’s efficacy is highly domain-dependent. In text-centric disciplines such as philosophy, law, and political science, its performance rivals that of state-of-the-art open-source reasoning models like DeepSeek-R1 and Qwen3-MoE. Conversely, in mathematics, physics, and chemistry, its results are statistically indistinguishable from random selection. Moreover, FMWMF is a pure answer-selection architecture; it offers no transparent rationale for why a given question–answer pair is matched and, unlike DeepSeek-R1, cannot articulate this alignment in natural language. An intriguing and forward-looking direction for future work is to embed the FMWMF framework directly within large language models. In particular, for complex problem-solving scenarios, integrating FMWMF with an attention mechanism

could direct the model's focus onto the most salient information points, offering a principled way to alleviate hallucinations, response incoherence, and training instabilities currently observed in LLMs.

Author Contributions: Methodology, Z.S.; Validation, Z.L.; Writing—original draft, J.H.; Writing—review & editing, X.H.; Supervision, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (62267003, 62177012).

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare that there are no conflicts of interest.

References

1. Amancio, L.; Dorneles, C.F.; Daniel, H.; Dalip, D.H. Recency and quality-based ranking question in CQAs: A Stack Overflow case study. *Inf. Process. Manag.* **2021**, *58*, 102552. [\[CrossRef\]](#)
2. Minaee, S.; Kalchbrenner, N.; Cambria, E. Deep Learning Based Text Classification: A Comprehensive Review. *ACM Comput. Surv.* **2020**, *54*, 1–40. [\[CrossRef\]](#)
3. Xu, J.; Zhang, W. Study of Intelligent Question Answering System Based on Ontology. In Proceedings of the 2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 16–18 October 2020; pp. 428–431. [\[CrossRef\]](#)
4. Wang, S.; Jiang, J. A compare-aggregate model for matching text sequences. *arXiv* **2016**, arXiv:1611.01747. [\[CrossRef\]](#)
5. Matthew, E.P.; Mark, N.; Mohit, I.; Matt, G. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 2–4 June 2018; Volume 1, pp. 2227–2237. [\[CrossRef\]](#)
6. Jacob, D.; Chang, M.-W.; Kenton, L. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186. [\[CrossRef\]](#)
7. Alec, R.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. Available online: <https://www.mikecaptain.com/resources/pdf/GPT-1.pdf> (accessed on 25 August 2025).
8. Tran, N.K.; Claudia, N. Multihop attention networks for question answer matching. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18), Ann Arbor, MI, USA, 8–12 July 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 325–334. [\[CrossRef\]](#)
9. Yang, Z.; Wang, Y.; Gan, J. Design and research of intelligent question-answering (Q&A) system based on high school course knowledge graph. *Mob. Netw. Appl.* **2021**, *26*, 1884–1890. [\[CrossRef\]](#)
10. Qian, Y. The Semantic Framework of Library Intelligent Question Answering System Based on Exploratory Search Behavior. In Proceedings of the 2022 IEEE 2nd International Conference on Computer Communication and Artificial Intelligence (CCAI), Beijing, China, 6–8 May 2022; pp. 65–70. [\[CrossRef\]](#)
11. Wildan, A.H.; Aji, R.F. Transformer and Large Language Models for Automatic Multiple-Choice Question Generation: A Systematic Literature Review. *IEEE Access* **2025**, *13*, 127100–127112. [\[CrossRef\]](#)
12. Hu, Z. Research and implementation of railway technical specification question answering system based on deep learning. In Proceedings of the 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 12–14 June 2020; pp. 5–9. [\[CrossRef\]](#)
13. Yu, L.; Hermann, K.M.; Blunsom, P.; Pulman, S. Deep learning for answer sentence selection. *arXiv* **2014**, arXiv:1412.1632. [\[CrossRef\]](#)
14. Vasin, P.; Roth, D.; Yih, W.-T. Mapping dependencies trees: An application to question answering. In Proceedings of the 8th International Symposium on Artificial Intelligence and Mathematics, Chongqing, China, 7–9 April 2023. Available online: <https://api.semanticscholar.org/CorpusID:8214465> (accessed on 25 August 2025).
15. Yih, S.W.-T.; Chang, M.-W.; Meek, C.; Pastusiak, A. Question answering using enhanced lexical semantic models. In Proceedings of the 51st Annual Meeting of the Association for Computational linguistics, Sofia, Bulgaria, 3 June 2013. Available online: <https://www.microsoft.com/en-us/research/publication/question-answering-using-enhanced-lexical-semantic-models/> (accessed on 25 August 2025).
16. Adebayo, I.M.; Kim, B.-S. Question-answering system powered by knowledge graph and generative pretrained transformer to support risk identification in tunnel projects. *J. Constr. Eng. Manag.* **2025**, *151*, 04024193. [\[CrossRef\]](#)

17. Di, W.; Eric, N. 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, Beijing, China, 26–31 July 2015; Volume 2, pp. 707–712. [\[CrossRef\]](#)
18. He, H.; Gimpel, K.; Lin, J. Multi-perspective sentence similarity modeling with convolutional neural networks. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 1576–1586. [\[CrossRef\]](#)
19. Wang, Z.; Hamza, W.; Florian, R. Bilateral multi-perspective matching for natural language sentences. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19 August 2017; pp. 4144–4150. [\[CrossRef\]](#)
20. Wiwin, S.; Pratama, R.A.; Rahadika, F.Y.; Purnomo, M.H.A. Self-Attention Mechanism of RoBERTa to Improve QAS for e-health Education. In Proceedings of the 2021 4th International Conference of Computer and Informatics Engineering (IC2IE), Depok, Indonesia, 14–15 September 2021; pp. 221–225. [\[CrossRef\]](#)
21. Guo, Y.; Zhang, J.; Chen, X.; Ji, X.; Wang, Y.-J.; Hu, Y.; Chen, J. Improving vision-language-action model with online reinforcement learning. *arXiv* **2025**, arXiv:2501.16664. [\[CrossRef\]](#)
22. Wang, R.; Zhang, Z.; Rossetto, L.; Ruosch, F.; Bernstein, A. NLQxform-UI: An Interactive and Intuitive Scholarly Question Answering System. In Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, Padua, Italy, 13–18 July 2025; pp. 3990–3993. [\[CrossRef\]](#)
23. Clark, K.; Luong, M.-T.; Le, Q.V.; Manning, C.D. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv* **2020**, arXiv:2003.10555. [\[CrossRef\]](#)
24. Laskar, M.T.R.; Huang, J.X.; Hoque, E. Contextualized embeddings based transformer encoder for sentence similarity modeling in answer selection task. In Proceedings of the Twelfth Language Resources and Evaluation Conference, Marseille, France, 11–16 May 2020; pp. 5505–5514. Available online: <https://aclanthology.org/2020.lrec-1.676/> (accessed on 25 August 2025).
25. Song, J.; Ashktorab, Z.; Pan, Q.; Dugan, C.; Geyer, W.; Malone, T.W. Interaction Configurations and Prompt Guidance in Conversational AI for Question Answering in Human-AI Teams. *arXiv* **2025**, arXiv:2505.01648. [\[CrossRef\]](#)
26. Zhang, Z.; Yang, J.; Zhao, H. Retrospective reader for machine reading comprehension. In Proceedings of the AAAI Conference on Artificial Intelligence, Singapore, 20–27 January 2021; Volume 35, pp. 14506–14514. [\[CrossRef\]](#)
27. Li, L.; Zhou, A.; Zhang, B.; Xiao, F. Multiple fragment-level interactive networks for answer selection. *Neurocomputing* **2020**, *420*, 80–88. [\[CrossRef\]](#)
28. Liu, C.; Jiang, S.; Yu, H.; Yu, D. Multi-turn Inference Matching Network for Natural Language Inference. In Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing, Hohhot, China, 26–30 August 2018; pp. 131–143. [\[CrossRef\]](#)
29. Guo, D.; Yang, D.; Zhang, H. Deepseek-r1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv* **2025**, arXiv:2501.12948. [\[CrossRef\]](#)
30. Wang, A.; Shu, D.; Wang, Y.; Ma, Y.; Du, M. Improving LLM Reasoning through Interpretable Role-Playing Steering. *arXiv* **2025**, arXiv:2506.07335. [\[CrossRef\]](#)
31. Hoffer, E.; Ailon, N. Deep metric learning using triplet network. In Proceedings of the Similarity-Based Pattern Recognition: Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, 12–14 October 2015; pp. 84–92. [\[CrossRef\]](#)
32. Wang, M.; Smith, N.A.; Mitamura, T. What is the Jeopardy model? A quasi-synchronous grammar for QA. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), Prague, Czech Republic, 28–30 June 2007; pp. 22–32. Available online: <https://aclanthology.org/D07-1003/> (accessed on 25 August 2025).
33. Yang, Y.; Yih, W.-T.; Meek, C. WikiQA: A Challenge Dataset for Open-Domain Question Answering. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 2013–2018. [\[CrossRef\]](#)
34. Gao, H.; Hu, M.; Cheng, R.; Gao, T. Hierarchical ranking for answer selection. *arXiv* **2021**, arXiv:2102.00677. [\[CrossRef\]](#)
35. He, J.; Zhang, H.; Hu, X. Open domain answer selection model fusing double matching-focus. *Comput. Eng.* **2023**, *49*, 303–310. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.