Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.060534

ARTICLE

Dialogue Relation Extraction Enhanced with Trigger: A Multi-Feature Filtering and Fusion Model

Haitao Wang[1,2], **Yuanzhao Guo**[1,2], **Xiaotong Han**[1,2] **and Yuan Tian**[1,2,*]

[1]School of Artificial Intelligence, Jilin University, Changchun, 130012, China

[2]Engineering Research Center of Knowledge-Driven Human-Machine Intelligence, MOE, Changchun, 130012, China

*Corresponding Author: Yuan Tian. Email: yuantian@jlu.edu.cn

Received: 04 November 2024; Accepted: 17 January 2025; Published: 26 March 2025

ABSTRACT: Relation extraction plays a crucial role in numerous downstream tasks. Dialogue relation extraction focuses on identifying relations between two arguments within a given dialogue. To tackle the problem of low information density in dialogues, methods based on trigger enhancement have been proposed, yielding positive results. However, trigger enhancement faces challenges, which cause suboptimal model performance. First, the proportion of annotated triggers is low in DialogRE. Second, feature representations of triggers and arguments often contain conflicting information. In this paper, we propose a novel **Multi-Feature Filtering and Fusion** trigger enhancement approach to overcome these limitations. We first obtain representations of arguments, and triggers that contain rich semantic information through attention and gate methods. Then, we design a feature filtering mechanism that eliminates conflicting features in the encoding of trigger prototype representations and their corresponding argument pairs. Additionally, we utilize large language models to create prompts based on Chain-of-Thought and In-context Learning for automated trigger extraction. Experiments show that our model increases the average F1 score by 1.3% in the dialogue relation extraction task. Ablation and case studies confirm the effectiveness of our model. Furthermore, the feature filtering method effectively integrates with other trigger enhancement models, enhancing overall performance and demonstrating its ability to resolve feature conflicts.

KEYWORDS: Dialogue relation extraction; feature filtering; chain-of-thought

1 Introduction

Relation extraction (RE) is a fundamental and important task [1,2] in information extraction, aimed at optimizing models to identify factual relations among arguments in unstructured text [3,4]. It has diverse applications, including knowledge graph construction [5,6], information retrieval [7,8], text summarization generation [3], sentiment classification [9,10], question answering [11], and more. These applications have significant impacts, ranging from enhancing search engine capabilities to powering intelligent assistants.

Among various RE tasks, dialogue relation extraction (DRE) aims to predict the relations between two arguments within a given dialogue, which is highly valuable for developing advanced dialogue systems [12]. DRE enhances conversational agents by improving their contextual understanding and intent recognition, which are essential for effective dialogue management [13,14]. Additionally, DRE provides critical insights into speaker interactions and dialogue semantics, contributing to research in computational social science and human-computer interaction.

With the remarkable success of pre-trained language models in natural language processing [15], these models have been tailored to address DRE tasks. This is the first type of DRE model, known as sequence-based models (e.g., BERT$_s$ [16], RoBERTa$_s$ [17], CoIn [18] and SimpleRE [19]), transforms dialogues into ultralong texts by adding special symbols and inputs them into a pre-trained language model for encoding and further relation classification. Recognizing the similarity between knowledge graph completion and DRE, graph-based models have been developed. These algorithms first construct a graph with linked elements from the dialogue, and use graph-based reasoning mechanisms to predict relations between arguments, incorporating token-based graphs [20], that consist of speaker, argument, utterance nodes [21], and even semantic networks [22] to capture interactions for DRE. Given the low information density in dialogues [23], where only a few meaningful phrases significantly aid DRE tasks, trigger-enhanced models have emerged to mine and utilize these critical elements, known as triggers, to improve DRE performance. This includes multi-task models, TREND [24], GRASP [25], and KEPT [26], which predict relations while simultaneously identifying triggers [27,28], whereas TLAG [29] concatenates the embeddings of predicted triggers with other learned representations for DRE.

While preliminary studies highlight the effectiveness of trigger-enhanced models, two key issues remain unaddressed. First, regarding trigger prediction, existing trigger-enhanced models have not addressed the issue of insufficient trigger annotations. Annotated triggers account for less than 30% of the samples in DialogRE. The lack of trigger annotations prevents the model from learning how to predict triggers and establish strong associations between triggers and target relations [30]. While triggers help narrow down relation categories, their scarcity limits the model's ability to learn associations effectively. Without enough triggers, the model is forced to rely on ambiguous information, which reduces its performance. Second, in terms of feature fusion, the learned representations of triggers and arguments often contain conflicting information, which is a problem that was not investigated and addressed in previous trigger-enhanced models. Triggers frequently associate with multiple relations, and arguments can have different relations across various pairs. This results in triggers and arguments having features that point to different relations, creating potential conflicts. If trigger and argument features are blindly concatenated for relation extraction, the model may struggle to determine the correct relation category for a sample. As shown in Fig. 1, the example includes the argument pair Emily and Speaker3, along with the corresponding trigger "honeymoon". The trigger "honeymoon" signifies two relations, per:girl/boyfriend and per:spouse, while Emily is associated with Speaker2 through the relation per:parents, it becomes challenging for the model to determine the correct relation category for the argument pair among per:girl/boyfriend, per:spouse, and per:parents, when using the concatenated representation of the trigger and argument pair.

To address the issues outlined above, we propose a novel trigger-enhanced model with **M**ulti-**F**eature **F**iltering and **F**usion (MF2F) for DRE. In MF2F, we implement a feature filtering mechanism to eliminate conflicting features in the trigger prototype representation and the encodings of the argument pair. The core technique used is average pooling. Inspired by the concept of class-center vectors, which retain consistent features while discarding inconsistencies, we employ the embeddings of the arguments to create a filtering template. This template undergoes an average pooling operation with the trigger's prototype representation. Additionally, we construct another filtering template using one argument's embedding and the trigger's prototype representation, which is then subjected to average pooling with the other argument's embedding. This process enables the model to retain the trigger and argument features that align with the same relation while filtering out their conflicting features. The filtered trigger representation and argument embeddings are then concatenated for relation classification. Furthermore, to mitigate the shortage of manually annotated triggers in the dataset, we leverage the powerful text understanding capabilities of large language models (LLMs) to automatically annotate triggers for the samples lacking manual annotations. We design prompts based on chain of thought (CoT) [31,32] and in-context learning (ICL) [33,34] to facilitate this automatic trigger annotation. In our experiments, our model achieves state-of-the-art (SOTA) performance on the benchmark dataset, significantly surpassing the strong baselines in previous works. Results from ablation studies, and availability validation further demonstrate the effectiveness, robustness, and scalability of the proposed feature filtering mechanism and LLM-based automatic trigger annotation. Beyond improving DRE performance, this work introduces a novel method for extracting triggers and presents a new perspective on their utilization.

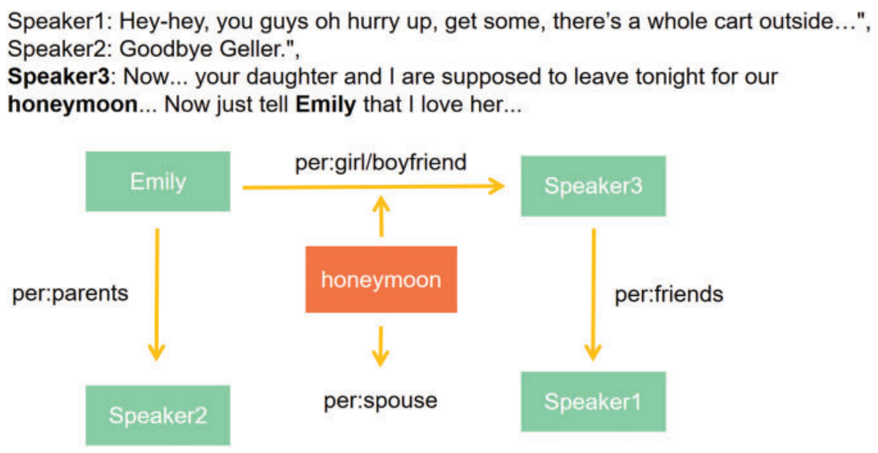

Speaker1: Hey-hey, you guys oh hurry up, get some, there's a whole cart outside…",
Speaker2: Goodbye Geller.",
Speaker3: Now... your daughter and I are supposed to leave tonight for our honeymoon... Now just tell Emily that I love her...
Emily
per:girl/boyfriend
Speaker3
honeymoon
per:parents
per:friends
Speaker2
per:spouse
Speaker1

Figure 1: An example of feature conflicts in triggers and arguments

Our contributions are as follows:

We introduce a novel DRE model, denoted as MF2F, which integrates an average-pooling-based feature filtering mechanism to eliminate conflicting features from the trigger prototype representation and the encodings of the given argument pair.

We utilize prompt tuning with an LLM to automatically annotate triggers for the samples lacking manual annotations, effectively addressing the challenge of insufficient manually annotated triggers in the dataset.

We conducted extensive experiments that demonstrate the proposed method achieves the SOTA performance in the DRE task while also validating its effectiveness, robustness, and scalability.

2 Related Work

DER models can be categorized into three types: sequence-based models, graph-based models, and trigger-enhanced models. Sequence-based models represent the earliest approach in DER, which later evolved into two branches: graph-based models and trigger-enhanced models.

Sequence-based models. These models concatenate the utterances, speakers, and special symbols into a long sentence, which is then input into a pre-trained language model for the DRE task. For instance, BERT$_s$ [16] uses BERT [35] as the encoder and adds special tokens to the input to indicate the start positions of the arguments. RoBERTa$_s$ [17] follows a similar input pattern as BERT$_s$, with the only modification being the replacement of the encoder with RoBERTa [36]. CoIn [18] introduces a mask mechanism along with a fusion gate and mutual attention to extract bi-grained embeddings for relation classification. SimpleRE [19] incorporates multiple [CLS] tokens in the input to capture various relations among different arguments and proposes a relation refinement gate to adaptively obtain relation-specific semantic embeddings. However, these models fail to adequately utilize the structural information present in dialogues, making them less effective.

Graph-based models. This category of methods transforms the sequential structure of a dialogue into a graph structure, framing the predicting of relations between arguments as a knowledge reasoning task. GDPNet [22] builds and refines a latent multi-view graph of tokens, whose representation is concatenated with the sequence-based representation for DRE. HGAT [21] employs graph neural networks to encode the relational information between arguments from a heterogeneous graph composed of linked speaker, argument, type, and utterance nodes. TUCORE-GCN [17] constructs a heterogeneous dialogue graph to capture interactions among dialogues, speakers, utterances and arguments. AMR [37] creates a sentence-level semantic network for each utterance, which are then interconnected to form a dialogue-level semantic network for modeling the entire dialogue. However, these methods do not adequately address the issue of low information density within dialogues.

Trigger-enhanced models. These methods increase the information density in dialogues by mining and utilizing information from triggers, thereby improving the performance of DRE models. They either conduct relation classification simultaneously with trigger prediction, or combine trigger representations with other learned representations. TREND [24] employs a multitask model to enhance relation classification through trigger identification. GRASP [25] (with an SOTA of 75.5) utilizes a prompt-based fine-tuning approach that performs mask-based relation prediction alongside argument and trigger detection. KEPT [26] introduces a DRE model that leverages the semantics of triggers and labels concurrently. TLAG [29] incorporates label-aware knowledge to guide the generation of trigger embedding, which are then integrated with other learned representations for DRE. However, two issues remain unresolved. First, in terms of trigger prediction, the proportion of annotated triggers in the dataset is low. Second, in feature fusion, the learned representations of triggers and arguments each contain conflicting features.

Unlike previous trigger-enhanced models that only relied on manually annotated triggers in the original dataset, we leverage LLMs to annotate triggers for unlabeled samples. This approach effectively mitigates the issue of trigger scarcity. Additionally, to address the feature conflict problem overlooked by existing trigger-enhanced models, we introduce a feature filtering mechanism. This not only improves the accuracy of relation extraction but also demonstrates the potential to scale DRE systems to more complex and dynamic conversational environments.

3 Proposed Methodology

3.1 Task Definition

In the DRE task, a dialogue d is defined as a tuple (S, U, A), comprising a set of speakers $S = \{s_k\}_{k=1}^{K}$, a set of utterances $U = \{u_n\}_{n=1}^{N}$ and a set of entity pairs $A = \{(e_i, e_j)_g\}_{g=1}^{G}$, where $u_n = \{s_k, x_{n,1}, x_{n,2}, ..., x_{n,M}\}$, $x_{n,m}$ refers to the m-th token in the n-th sentence spoken by the k-th speaker s_k, and e_i and e_j refer to the subject and object of the g-th entity pair, respectively. Given an argument pair (e_i, e_j) within the dialogue d, the goal of DRE is to predict the relations $r_{i,j} \in R$ between e_i and e_j, where R refers to the relation set of the dataset. Additionally, the datasets often include several useful words from the sentences as triggers to assist in relation prediction for specific argument pairs.

3.2 Model Overview

To effectively filter out conflicting features and further fuse the consistent features of the arguments and triggers, we propose the MF2F model. This model comprises five modules that form a cohesive pipeline, as illustrated in Fig. 2.

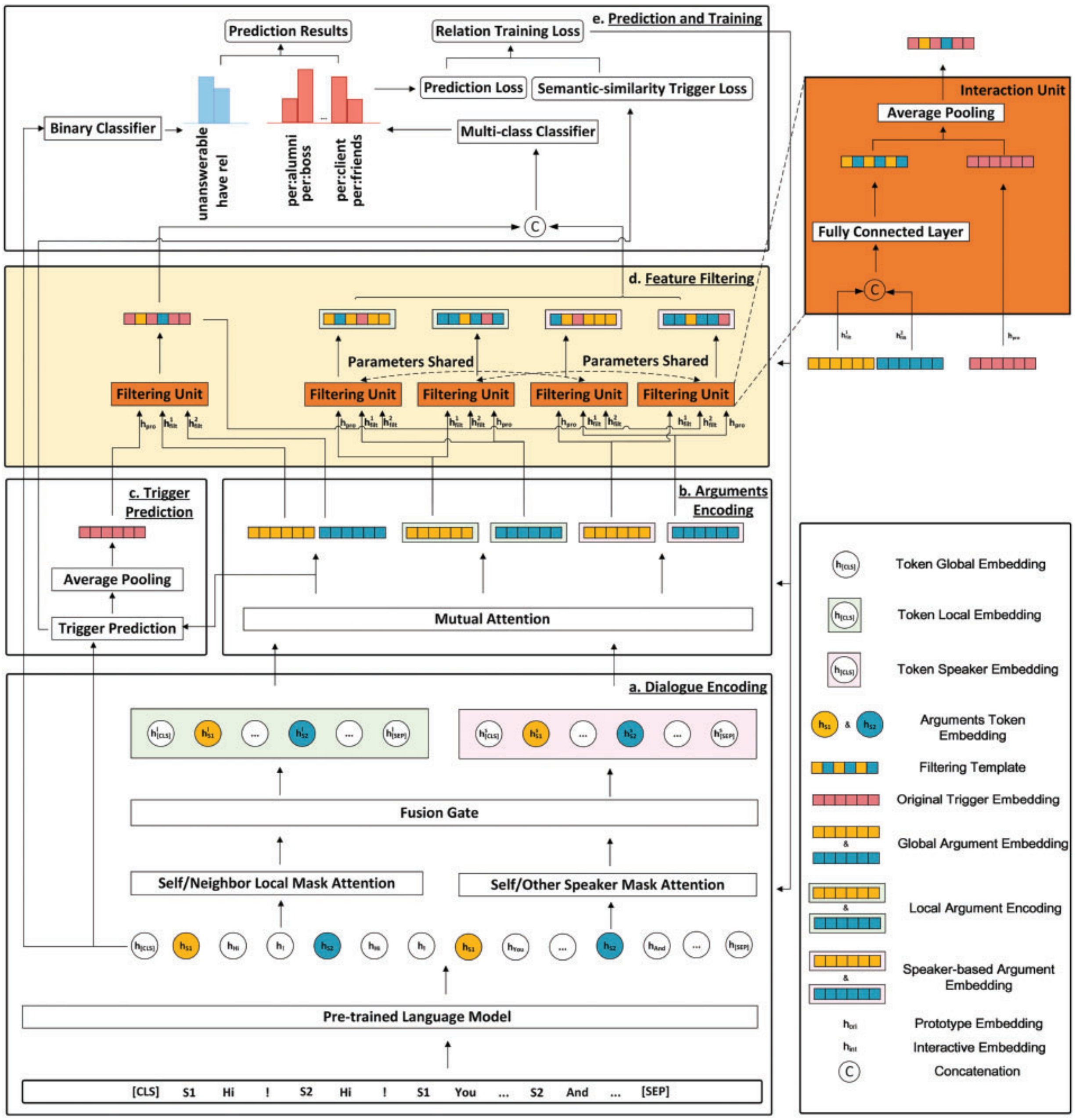


Figure 2: The model architecture of MF2F. MF2F consists of five modules: a) Dialogue Encoding module includes pre-trained model encoding, local and speaker mask attention mechanisms, and fusion gate structures; b) Arguments Encoding module incorporates the mutual attention mechanism; c) Trigger Prediction module has trigger prediction and trigger pooling methods; d) Feature Filtering module includes feature filtering units designed to mitigate the feature conflict issue; e) Prediction and Training module encompasses relation prediction and training loss calculation

The first module is the dialogue encoding module, which utilizes a pre-trained language model with the mask-based multihead self-attention mechanism to obtain three scales of dialogue token embeddings. The second module is the argument encoding module, where an attention mechanism aggregates different mentions to form the argument embedding. Using the three scales of dialogue token embeddings, we derive three corresponding argument embeddings. The third module is the trigger prediction module, which includes a discriminator followed by an average pooling layer to identify trigger tokens and generate their original prototype embeddings for a given argument pair. The fourth module is the feature filtering module, which constructs a filter unit to eliminate conflicting features between the argument and the trigger. The filtering unit primarily utilizes average pooling to preserve consistent features between the original representation and the filtering template while discarding inconsistent features, which are likely the culprits of conflict. Specifically, for a trigger and its corresponding two arguments, when one of the three requires feature filtering, the representations of the other two are aggregated into a template via a fully connected layer. Finally, in the prediction and training module, the filtered trigger and argument representations are combined and fed into a relation classifier for RE.

We then incorporate a relation classification loss alongside the trigger prediction loss to optimize the model, transforming our MF2F into a multitask model and further enhancing its performance in the DRE task. Specifically, before training, we utilize prompt tuning with an LLM to automate the annotation of triggers for samples lacking manual annotations, thereby addressing the issue of insufficient annotated triggers in the dataset.

3.3 Dialogue Encoding

In this section, we obtain three types of dialogue encodings: global, local, and speaker-based. The first two encodings capture dependencies between tokens at the dialogue and utterance levels, respectively, while the third focuses on token dependencies based on whether the utterances are spoken by the same person.

First, we concatenate the utterances and special tokens into a long sequence X:

$$X = [[\text{CLS}], x_{1,1}, \ldots, x_{1,M_1}, \ldots, x_{N,1}, \ldots, x_{N,M_N}, [\text{SEP}]] \tag{1}$$

where [CLS] and [SEP] serve as special tokens indicating the classification and the end of the dialogue, respectively. This dialogue X is then fed into a pre-trained model [38,39], such as RoBERTa, to obtain the token embeddings H:

$$H = RoBERTa(X) = [h'_{[\text{CLS}]}, h'_{x_{1,1}}, \ldots, h'_{x_{1,M_1}}, \ldots, h'_{x_{N,1}}, \ldots, h'_{x_{N,M_N}}, h'_{[\text{SEP}]}] \tag{2}$$

Next, we add the sinusoid position information H_{pos} and argument type information H_{type} to H to generate the global token embeddings H_g:

$$H_g = H + H_{pos} + H_{type} = [h_{[\text{CLS}]}, h_{x_{1,1}}, \ldots, h_{x_{1,M_1}}, \ldots, h_{x_{N,1}}, \ldots, h_{x_{N,M_N}}, h_{[\text{SEP}]}] \tag{3}$$

To capture local interactions between a token and others within the same utterance or neighboring utterances, we design a local mask within the multihead self-attention mechanism [40]:

$$H_q = \left[\text{head}_1^q; \ldots, \text{head}_i^q; \ldots, \text{head}_{N_a}^q\right] \tag{4}$$

$$\text{head}_i^q = \text{softmax}\left(\frac{H_g W_i^Q \cdot \left(H_g W_i^K\right)^T}{\sqrt{d_k}} + M_q\right) H_g \tag{5}$$

where $W^Q \in \mathbb{R}^{d \times d_q}$, and $W^K \in \mathbb{R}^{d \times d_k}$ are trainable parameters, and M_q represents the local mask. There are two types of local masks: one is the self-local mask M_{sl}, that is defined as:

$$M_{sl}[i, j] = \begin{cases} 0, & \text{if the } i\text{th and the } j\text{th token belongs to the same utterance} \\ -\infty, & \text{otherwise} \end{cases} \tag{6}$$

The other is the neighbor-local mask M_{nl}, that is defined as:

$$M_{nl}[i,j] = \begin{cases} 0, & \text{if the } i\text{th and the } j\text{th token belongs to the adjacent utterance} \\ -\infty, & \text{otherwise} \end{cases} \tag{7}$$

We calculate self-local token embeddings H_{sl} and neighbor-local token embeddings H_{nl} using Eqs. (4) and (5) with respective local masks. These embeddings are then fused together to generate the local token embeddings H_l using a gate mechanism:

$$H_l = G_l \circ H_{sl} + (1 - G_l) H_{nl} \tag{8}$$

$$G_l = \text{sigmoid}(FC(I_l)) \tag{9}$$

where $I_l = [H_{sl}; H_{nl}; H_{sl} - H_{nl}; H_{sl} \circ H_{nl}]$; and $FC()$ represents a fully connected layer.

Because the speaker information is particularly relevant in dialogue encoding, we employ two speaker-based masks: the self-speaker mask M_{ss} [41] and the other-speaker mask M_{os} as defined in Eqs. (4) and (5), to obtain embedding H_{ss} and H_{os}, respectively. The two speaker masks are defined as:

$$M_{ss}[i,j] = \begin{cases} 0, & \text{if the } i\text{th and the } j\text{th token is spoken by the same speaker} \\ -\infty, & \text{otherwise} \end{cases} \tag{10}$$

$$M_{os}[i,j] = -\infty - M_{ss}[i,j] \tag{11}$$

Finally, self-speaker token embeddings H_{ss} and other-speaker token embeddings H_{os} are blended into the speaker-based token embeddings H_s using a gate mechanism similar to that in Eqs. (8) and (9).

3.4 Argument Encoding

After obtaining different scales of dialogue encodings, we use them to calculate three types of argument encodings: global argument encoding, local argument encoding, and speaker-based argument encoding. While these encodings follow the same calculation process, they differ in the type of dialogue encoding they utilize.

To capture the various dependencies between an argument and the other mentions, we apply the mutual attention mechanism to obtain the global encoding $h^g_{e_i}$, the local encoding $h^l_{e_i}$ and the speaker-based encoding $h^s_{e_i}$ for the argument e_i. Let $h^q_{e_i}$ represent a specific embedding of the argument e_i, where $q \in \{g, l, s\}$. It is defined as follows:

$$h^q_{e_i} = \sum_{m=1}^{N_i} \alpha_{j,i,m} \cdot h^q_{i,m} \tag{12}$$

where N_i represents the number of mentions for the argument e_i, and $h^q_{i,m}$ represents the embedding of the m-th mention of e_i. This is calculated as follows:

$$h^q_{i,m} = \frac{1}{b_m - c_m + 1} \sum_{z=b_m}^{c_m} h^q_z \tag{13}$$

where h^q_z represents the token embedding corresponding to the j-th mention of e_i; and z is the index of the token in the dialogue encoding, ranging from the start b_m to the end c_m. $\alpha_{j,i,m}$ represents the attention weight between $h^q_{i,m}$ and the embedding of the other argument e_j in a given argument pair. It is defined as:

$$\alpha_{j,i,m} = \frac{exp\left(h'^{q}_{e_j} \cdot h^{q}_{i,m}\right)}{\sum_{n=1}^{N_i} exp\left(h'^{q}_{e_j} \cdot h^{q}_{i,m}\right)} \tag{14}$$

$$h'^{q}_{e_j} = \frac{1}{N_j} \sum_{m=1}^{N_j} h^{q}_{j,m} \tag{15}$$

where $h^{q}_{j,m}$ represents the embedding of the m-th mention of e_j.

3.5 *Trigger Prediction*

Based on previous studies [16,42], most triggers are notional words, including nouns, verbs, and adjectives. We collect noun, verb, and adjective tokens as candidate triggers. A discriminator is then constructed to calculate the probability P of each candidate c being a trigger token for a given entity pair e_i and e_j. The probability P is calculated as follows:

$$p\left(c \left| e_i, e_j\right.\right) = \text{sigmoid}\left(FC\left(h'_c\right)\right) \tag{16}$$

$$h'_c = \left[h^g_c; h^g_{e_i}; h^g_{e_j}\right] \tag{17}$$

where $h^g_{e_i}$ and $h^g_{e_j}$ represent the global embeddings of e_i and e_j, respectively, and h^g_c represents the global embedding of c. We select the top-K_t candidates with the highest probabilities as the true triggers, which are then used to obtain their original prototype trigger embedding $h^t_{e_i,e_j}$ using an average pooling layer as:

$$h^t_{e_i,e_j} = \frac{1}{K_t} \sum_{k=1}^{K_t} h^g_k \tag{18}$$

where h^g_p represents the global embedding of the k-th final predicted trigger token.

3.6 *Feature Filtering*

Before combining the embeddings of the trigger with those of the arguments for relation classification, it is essential to filter out features that may mutually influence each other. On the one hand, because multiple relations often share the same triggers, the features of a trigger carry semantic information from various relations. On the other hand, an argument may belong to more than one relation, meaning its features also encompass semantic information from diverse relations. This overlap can lead to incorrect classification outcomes if the full features of the trigger and argument pair are used. To address this issue, we propose a filtering unit. Its core technique is average pooling, which retains consistent features from the original vectors while filtering out inconsistent ones. Given a trigger t and two arguments e_i and e_j, the feature filtering process for t is defined as follows:

$$\zeta^t_{e_i,e_j} = \frac{1}{2}\left(h^t_{e_i,e_j} + h^g_{e_i,e_j}\right) \tag{19}$$

$$h^g_{e_i,e_j} = FC\left(\left[h^g_{e_i}; h^g_{e_j}\right]\right) \tag{20}$$

where $h^t_{e_i,e_j}$ represents the prototype embedding of t; $h^g_{e_i}$ and $h^g_{e_j}$ represent the global embedding of e_i and e_j, respectively; and $h^g_{e_i,e_j}$ represents the filtering template generated by $h^g_{e_i}$ and $h^g_{e_j}$. Similarly, the feature filtering process for e_i and e_j is defined as follows:

$$\zeta^q_{e_i} = \frac{1}{2}\left(h^q_{e_i} + h^q_{e_j,t}\right) \tag{21}$$

$$h^q_{e_j,t} = FC\left([h^g_{e_j}; \zeta^t_{e_i,e_j}]\right) \tag{22}$$

where $h^q_{e_j}$ represents one of the local embedding and speaker-based embedding for the argument e_j, i.e., $q \in \{l, s\}$; and $h^q_{e_j,t}$ represents the filtering template created by $h^q_{e_i}$ and $\zeta^t_{e_i,e_j}$.

3.7 Prediction and Training

We combine the results of relation existence prediction and relation classification to make the final prediction.

For relation existence, we use a binary classifier to evaluate the relation between the two given arguments e_i and e_j:

$$flag_{exist} = \begin{cases} \text{True} & \text{if } P_{exist} \geq 0.5 \\ \text{False} & \text{otherwise} \end{cases} \tag{23}$$

$$P_{exist} = \text{Binary Classifier}(X) \tag{24}$$

where X represents the dialogue tokens sequence.

For relation classification, we concatenate the filtered embeddings and input them into a multiclass classifier to predict relations for the argument pair:

$$\hat{r}_{class} = \{i \in \{1, \ldots, r, \ldots, |R|\}, p_i \geq 0.5\} \tag{25}$$

$$P = \left[p_1, \ldots, p_r, \ldots, p_{|R|}\right] = \text{sigmoid}(FC(H_{con})) \tag{26}$$

$$H_{con} = \left[\zeta^l_{e_i}; \zeta^l_{e_j}; \zeta^s_{e_i}; \zeta^s_{e_j}; \zeta^t_{e_i,e_j}\right] \tag{27}$$

where $\hat{r}_{class}$ represents the predicted relations for relation classification; p_r denotes the probability of the argument pair belonging to the r-th relation; $\zeta^t_{e_i,e_j}$ is the filtered embedding of trigger t; and ζ^q_e represents one of the two embeddings of argument e (i.e., $q \in \{l, s\}$).

The final RE result is obtained as follows:

$$\hat{r} = \begin{cases} \hat{r}_{class} & \text{if} flag_{exist} = \text{True and } \hat{\text{r}}_{\text{class}} \neq \phi \\ \text{unanswerable} & \text{otherwise} \end{cases} \tag{28}$$

where "unanswerable" represents that the argument pair has no relation.

We then adopt both relation classification loss and trigger prediction loss to optimize the model, transforming our MF2F into a multitask model, which enhances performance in the DRE task. The final loss of our model is defined as follows:

$$loss = \alpha \cdot loss_{rel} + (1 - \alpha) \cdot loss_{tri} \tag{29}$$

where α is a hyperparameter that regulates the balance between the two losses; and $loss_{rel}$ and $loss_{tri}$ represent the relation classification loss and the trigger prediction loss, respectively, which is defined as:

$$loss_{rel} = -\frac{1}{|A|} \sum_{(e_i,e_j) \in A} \sum_{r \in R} \left[y_{e_i,e_j,r} \log p_{e_i,e_j,r} + \left(1 - y_{e_i,e_j,r}\right)\left(1 - \log p_{e_i,e_j,r}\right)\right] \tag{30}$$

$$loss_{tri} = -\frac{1}{|A|} \sum_{(e_i,e_j) \in A} \frac{1}{\left|T^{can}_{e_i,e_j}\right|} \sum_{\hat{t} \in T^{pre}_{e_i,e_j}} w_{\hat{t}} \left(y^{\hat{t}}_{e_i,e_j} \log p^{\hat{t}}_{e_i,e_j} + \left(1 - y^{\hat{t}}_{e_i,e_j}\right)\left(1 - \log p^{\hat{t}}_{e_i,e_j}\right)\right) \tag{31}$$

where $T^{pre}_{e_i,e_j}$ denotes the set of predicted trigger tokens for the given arguments e_i and e_j; $y^{\hat{t}}_{e_i,e_j}$ represents the true label for the token $\hat{t}$; $p^{\hat{t}}_{e_i,e_j} = p\left(\hat{t}\,|e_i, e_j\right)$ calculated by Eq. (15); and $w_{\hat{t}}$ represents a weight that encourages the model to identify tokens with similar semantics to the trigger tokens as supplements. It is defined as follows:

$$w_{\hat{t}} = \begin{cases} 1 - score_{\hat{t}}, & \text{if } score_{\hat{t}} > score_{th} \\ 1 & \text{otherwise} \end{cases} \tag{32}$$

$$score_{\hat{t}} = \max_{t \in T^{tru}_{e_i,e_j}} \cos\left(\hat{t}, t\right) \tag{33}$$

where $T^{tru}_{e_i,e_j}$ represents the set of true trigger tokens for two given arguments e_i and e_j; and $score_{th}$ represents a threshold that determines whether the semantics between $\hat{t}$ and t is close enough with SBert [43].

Before training, we leverage the strong text comprehension abilities of an LLM to automatically annotate triggers for samples lacking manual annotations. We design prompts based on CoT and ICL to facilitate this task. In constructing the CoT, we break down the trigger annotation process into three subtasks. First, we prompt the LLM to explain why a given argument pair has such a particular relation, encouraging the model to explore the semantics relevant to the task. Second, we prompt the LLM to predict the trigger and assess its accuracy. Third, we prompt the LLM to provide an explanation for each predicted trigger. The prompt also includes a contextual example to help the LLM learn the thought patterns necessary for trigger identification and to master the correct output format. For quality control in predicting triggers [44], we employ the Local Outlier Factor (LOF) [45] to evaluate the quality of the explanations. Outliers are considered poor explanations, and the corresponding triggers are discarded to ensure the acquisition of high-quality triggers.

4 Experiments

4.1 Datasets

The dataset used in our experiments is the DialogRE dataset, which comprises dialogues extracted from the American TV series *Friends*. It includes two versions: DialogREv1 and DialogREv2. DialogRE contains 1788 dialogues, 37 relations, and 8119 relational triplets, the majority of which describe the relations between characters in the dialogues. Notably, annotated triggers are provided for a subset of these triplets. For our experiments, we utilize the standard three partitions of the data: training, development, and testing, as structured in the DialogRE dataset.

4.2 Baselines and Evaluation Metrics

To conduct a comprehensive performance evaluation, we compare our model against 11 baselines and SOTA methods, which are categorized into three groups, including **Sequence-based models:** BERT_s [16], RoBERTa_s [17], CoIn [18] and SimpleRE [19]; **Graph-based models:** HGAT [21], GDPNet [22], AMR [37] and TUCORE-GCN [17]; and **Trigger-enhanced models:** TREND [24], GRASP [25], TLAG [29] and KEPT [26]. We use the standard Precision (P), Recall (R) and micro-F1 (F1) to evaluate the model performance:

$$\text{P} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{34}$$

$$\text{R} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{35}$$

$$\text{F1} = \frac{2 * \text{P} * \text{R}}{\text{P} + \text{R}} \tag{36}$$

where TP, FP and FN stand for true positive, false positive, and false negative, respectively. In the rest of the paper, F1 refers to micro-F1 unless otherwise specified.

4.3 Implementation Details

Our model utilizes AdamW as the optimizer with a Cosine Annealing scheduler, with a weight decay of 1e-3. The pre-trained model RoBERTa-large serves as our encoder, with a learning rate of 5e-6. We trained the model using a batch size of 2 for 30 epochs, with a learning rate of 1e-4 for parameters other than those of the pre-trained model. The multitask learning loss weights are assigned as 0.90 for RE and 0.10 for trigger prediction. We insert special tokens to represent speaker indices between the utterances, forming an input sequence. This sequence is then divided into sub-word tokens. If the length of the tokenized sequence exceeds 512, we split it into two overlapping sub-sequences. Experiments were conducted on a server equipped with two NVIDIA TITAN RTX GPUs, while the environment for LLM-based trigger annotation experiments was based on an Intel Core i7-12700K processor.

4.4 Overall Results

As shown in Table 1, our MF2F achieves the best performance among all models, reaching an F1 score of 76.3% on DialogREv1 and 77.0% on DialogREv2. It surpasses the best baseline, GRASP, by 1.2% and 1.5% on both versions of the DialogRE dataset, respectively. This clearly demonstrates the effectiveness of our approach.

Table 1: The F1 (%) on DialogREv1 and DialogREv2 test datasets. The best results are bold

Type	Models	DialogREv1	DialogREv2
Sequence-based	BERT_s [16]	61.2	59.5
	RoBERTa_s [17]	–	71.3
	CoIn [18]	72.3	–
	SimpleRE [19]	66.3	66.7
Graph-based	HGAT [21]	–	56.1
	GDPNet [22]	64.9	60.2
	AMR [37]	67.3	67.1
	TUCORE-GCN [17]	–	73.1
Trigger-enhanced	TREND [24]	–	67.8
	GRASP [25]	75.1	75.5
	TLAG [29]	–	66.6
	KEPT [26]	–	73.6
	MF2F	**76.3 (↑ 1.2)**	**77.0 (↑ 1.5)**

Trigger-enhanced models outperform the other two categories. Compared to sequence-based and graph-based models, trigger-enhanced models achieve an average improvement of at least 2.3% in F1 score on the test dataset. This confirms the efficacy of enhancing the DRE model with triggers through multitask learning and feature filtration.

Moreover, our model outperforms all other trigger-enhanced models. In addition to multitask learning and feature filtration, it incorporates feature filtering and LLM-based automatic trigger annotation. These mechanisms further enhance the model’s performance in the DRE task.

4.5 Ablation Study

To showcase the efficacy of the feature filtering mechanism, we established two ablation scenarios. The first scenario, labeled $\text{MF2F}_{\text{rdm\_tpl}}$, uses filtering templates from randomly initialized vector, replacing each of the three filter templates with random templates individually and calculating the average results. The second scenario, denoted as $\text{MF2F}_{\text{wot\_tpl}}$, utilizes the original embeddings of the arguments and the trigger prototype directly for relation classification, bypassing feature filtering.

As indicated in Table 2, we observe an average drop of 3.2% in the F1 score across the dataset when the feature filtering step is removed, underscoring the effectiveness of this mechanism. Interestingly, even when using a randomly initialized vector to create the filtering template, the model still achieves over 0.8% improvement compared to the scenario without feature filtering, although this does not match the performance of templates derived from the argument or trigger. This suggests that the model can learn a relatively effective filtering template from scratch through optimization, further confirming the utility of the feature filtering mechanism from a different perspective.

Table 2: The results of ablation experiments

Model	**DialogREv1**			**DialogREv2**		
	P	**R**	**F1**	**P**	**R**	**F1**
MF2F	78.7	74.1	76.3	79.2	75.0	77.0
$\text{MF2F}_{\text{rdm\_tpl}}$	75.5	73.0	74.2	75.2	73.2	74.3
$\text{MF2F}_{\text{wot\_tpl}}$	77.1	70.2	73.5	75.7	71.2	73.4
$\text{MF2F}_{\text{wot\_aua}}$	75.9	72.8	74.3	77.5	72.9	75.2
$\text{MF2F}_{\text{ful\_aua}}$	77.3	72.0	74.6	78.9	73.2	75.9

We also developed two additional ablation scenarios to assess the effectiveness of our LLM-based automatic trigger annotation strategy. In the first scenario, $\text{MF2F}_{\text{wot\_aua}}$, triggers automatically annotated by the LLM are excluded from the training phase. In the second scenario, $\text{MF2F}_{\text{ful\_aua}}$, all triggers used in the training phase are obtained through automatic annotation by the LLM.

The results presented in Table 2 show an average performance reduction of 1.9% when triggers automatically annotated by the LLM are excluded. Furthermore, the model's performance drops by 1.4% on average when the triggers are entirely annotated by the LLM; however, it still outperforms the case without any automatically annotated triggers. These findings suggest that LLM-based automatic trigger annotation based is a valid approach for improving model performance in the DRE task.

4.6 Loss Weight Study

In trigger-based dialogue relation extraction, the loss function weight is a critical hyperparameter that balances the losses of the two tasks during training. It ensures that while relation extraction remains the primary focus, the trigger prediction task is also optimized. To investigate its impact, we conducted experiments with α values of 0.80, 0.85, 0.90, 0.95, and 1.00 on DialogREv2, while keeping all other parameters constant.

As shown in Fig. 3, experimental results indicate that an excessive or insufficient focus on the relation extraction task negatively impacts performance. When training exclusively focuses on the relation extraction task, the F1 drops to its lowest value of 75.0%.

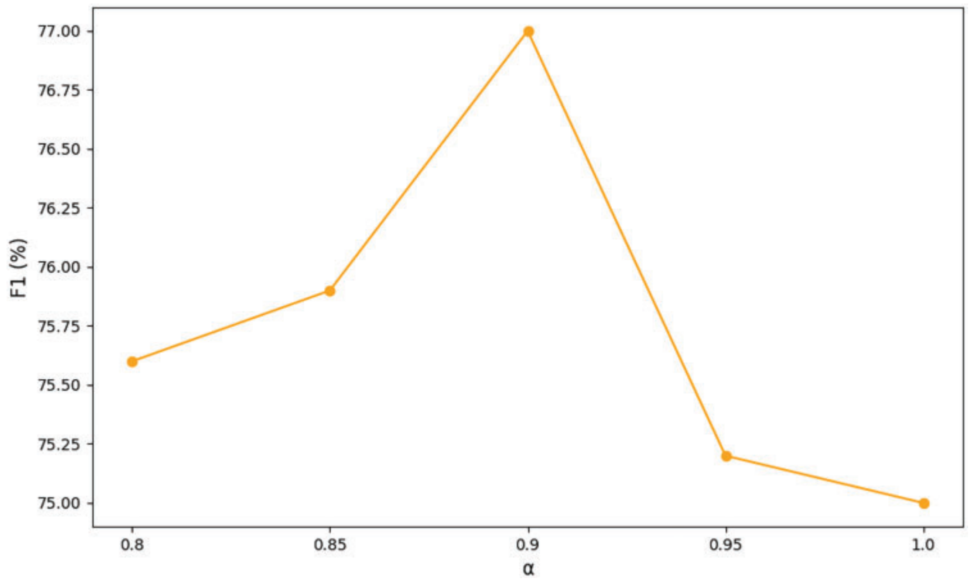


Figure 3: The results of the loss weight study

4.7 Availability Validation

To evaluate the applicability of the two innovative technologies in our model, we integrated the feature filtering and LLM-based trigger annotation into the recent trigger-enhanced DRE model, TREND. The results are showcased in Table 3. By incorporating the feature filtering into TREND, its performance increases by 4.7%. When applying the LLM-based trigger annotation, TREND's performance improves by 1.7%. Notably, when both technologies are implemented simultaneously, the performance is enhanced by 5.2%. These findings suggest that both feature filtering and LLM-based trigger annotation can be utilized individually or together in other DRE models to effectively improve their performance.

Table 3: Performance of TRNED using feature filtering and LLM annotating

Model	DialogREv2			DialogREv2+LLM annotating		
	P	**R**	**F1**	**P**	**R**	**F1**
TRNED	69.7	69.4	69.6	71.4	71.2	71.3
TRNED+filter	74.0	74.6	74.3	74.7	75.0	74.8

4.8 Case Study

To demonstrate the working principle of the large model data augmentation method and the feature filtering mechanism in MF2F, we conducted four case studies on the DialogREv2 test dataset. In the Case1, we focus on investigating whether LLMs can extract appropriate trigger words to enhance the training of our MF2F. In the Case2, we focused on scenarios where a single trigger corresponds to two relations, using MF2Fwot$_t$pl (without filtering templates) and MF2F (with filtering templates) to perform relation prediction on the test set, and visualized the representation space of the relation using principal component analysis (PCA). In the Case3, we focused on scenarios where a single entity corresponds to two relations, using MF2Fwot$_t$pl and MF2F to perform relation prediction and also visualized the representation space of the relation. In the Case4, to further investigate the capability of MF2F under complex conditions, we divided DialogREv2 based on the number of arguments, and compared the results of MF2F and GRASP under these conditions.

As shown in Table 4, we selected three examples that originally did not have annotation triggers to illustrate the impact of these annotations extracted by LLMs. In Dialogue1, Speaker2 fired Tim, with the LLM extracting the trigger "fire", directly indicating the boss-employee relation. In Dialogue2, Speaker1 visited Vail, and the LLM's extraction of "go skiing" suggests that Speaker1 skied during the visit. Dialogue3 follows a similar pattern. These examples highlight how LLM annotations compensate for the lack of human annotations, effectively training the model and thereby enhancing its performance.

Table 4: LLM annotation examples of the Case1

Id	Dialogue	Arguments	Relation	Trigger
1	Speaker1: I had the best time with Tim... Speaker2: I... I have to **fire** him.	Speaker2 Tim	per:subordinate	fire
2	Speaker1:... I can spend Thanksgiving with my family. See, every year we **go skiing** in Vail...	Speaker1 Vail	per:visited_place	go skiing
3	Speaker1: Oh! Okay, **call me**! Speaker 4: Okay, look, I-I know what...	Speaker1 Speaker4	per:dates	call me

In the Case2 and Case3, as shown in Fig. 4, we observed that without filtering templates, F1 scores dropped by 19.8% and 9.3%, underscoring the effectiveness of the filtering mechanism in both scenarios.

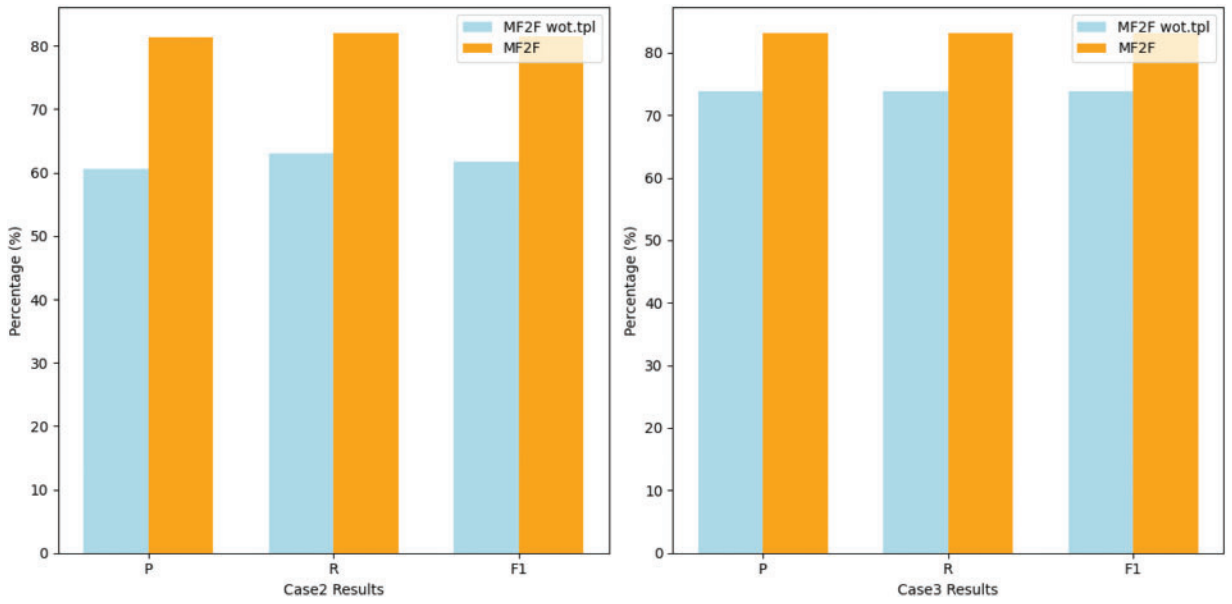
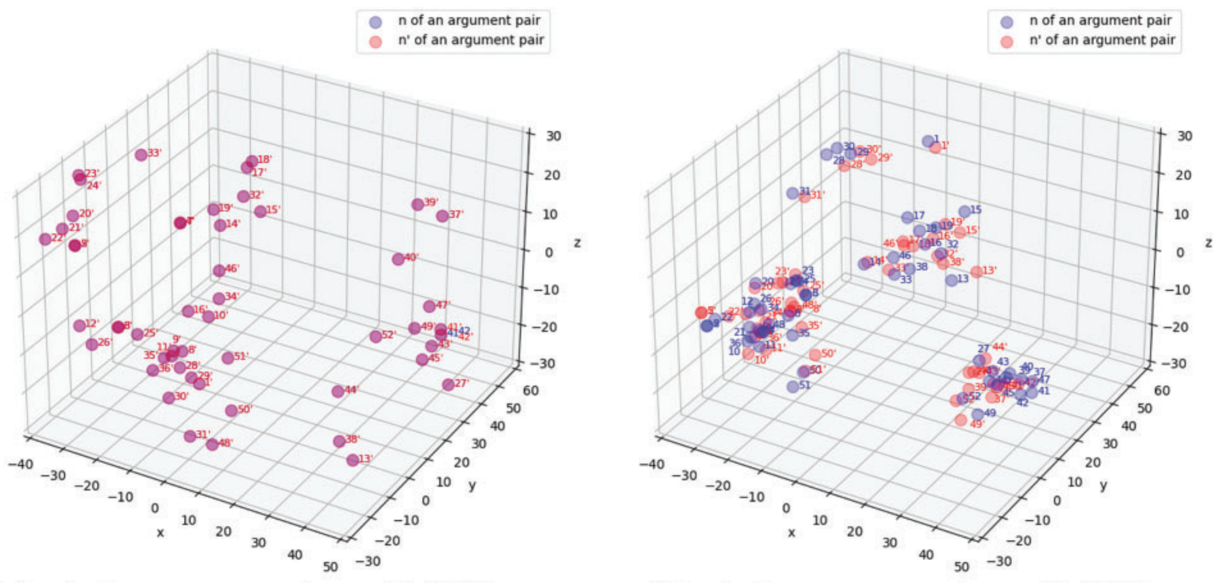


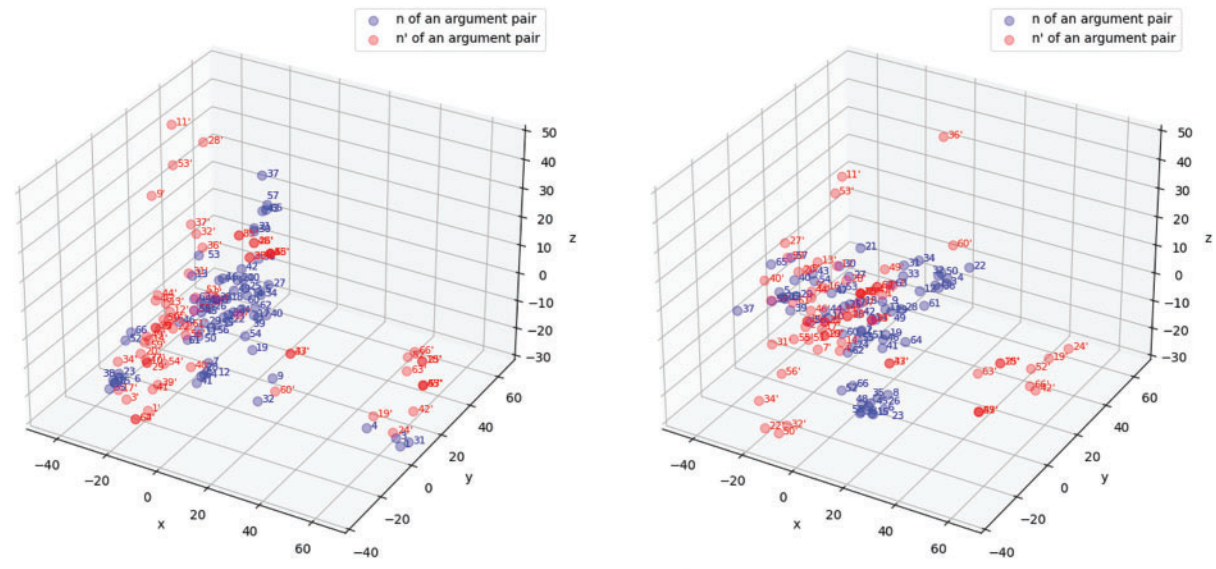
Figure 4: The results of the Case2 and Case3 on DialogREv2 datasets. In the Case2, a single trigger corresponds to two relations. In the Case3, a single entity in the test set belongs to two relations in a dialogue

Figs. 5 and 6 illustrate that for pairs of triples corresponding to the same trigger or the same entity, the red dots represent relation representation for one triple, while the blue dots represent those for the other, easily confusable triple. The filtering mechanism successfully separated previously overlapping relation representations, resulting in clearer boundaries and enabling effective classification of triples with different relations under similar conditions.



(a) relation representations of MF2F$_{\text{wot\_tpl}}$ (b) relation representations of MF2F

Figure 5: The results of PCA visualization of the relation representations in the Case2



(a) relation representations of MF2F$_{\text{wot\_tpl}}$ (b) relation representations of MF2F

Figure 6: The results of PCA visualization of the relation representations in the Case3

The results indicate that while MF2F performs slightly worse than GRASP by 0.9% on data with a small number of arguments, it significantly outperforms GRASP on data with a larger number of arguments, as shown in Fig. 7. For data with a small number of arguments, MF2F still has room for improvement, as the limited number of conflicts restricts the model's ability to fully demonstrate its advantages. In contrast, in dialogues with a larger number of arguments, where conflicts occur more frequently, the model effectively handles these challenges. This demonstrates the efficiency of our feature filtering method in mitigating feature conflict issues.

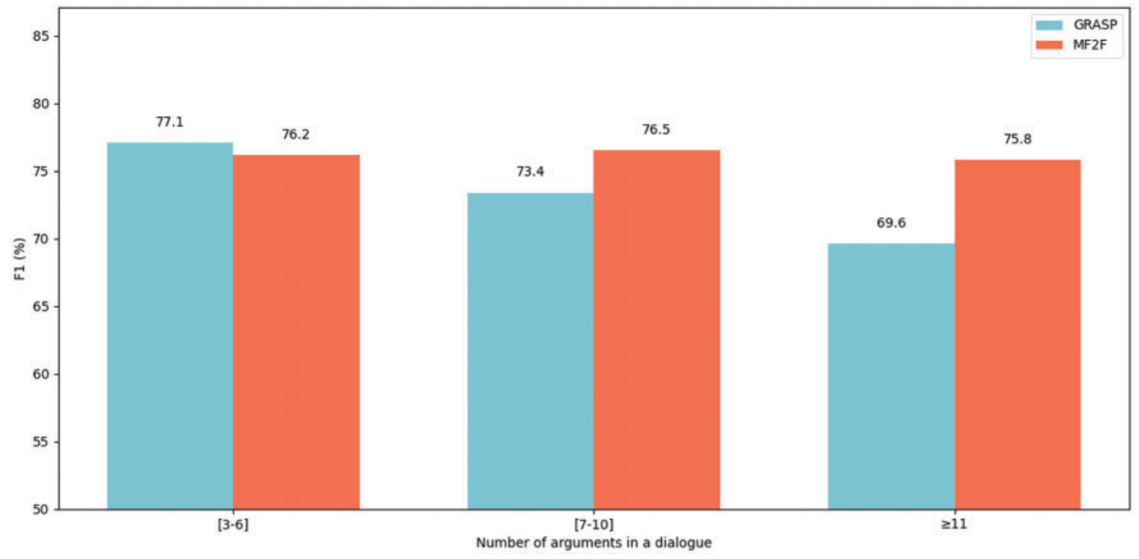


Figure 7: The results of the Case4

5 Conclusion

In this paper, we introduce the MF2F model, which implements two innovative techniques: automatic LLM-based trigger annotation and average pooling-based feature filtering. These techniques aim to address issues of insufficient trigger annotations and conflicting information in trigger and argument representations for the dialogue relation extraction task. We employ prompt tuning of an LLM to achieve automatic trigger annotation for samples lacking manual annotations. The prompt construction follows a CoT structure, and includes contextual examples for the LLM to learn from. In the feature filtering stage, average pooling allows the model to retain features that point to the same relations while eliminating individual features that indicate other incorrect relations. This enhances the discriminability of a sample concerning its correct relation. Experiment results indicate that trigger-enhanced models outperform sequence-based and graph-based models, with our MF2F achieving the best performance among trigger-enhanced models. Through ablation studies, we validated the effectiveness of our two innovative techniques. Furthermore, the innovative techniques integrated into MF2F can be easily applied to other strong trigger-enhanced models, improving their performance in DRE tasks.

Acknowledgement: We would like to express our sincere gratitude to all the teachers and students involved for their support and collaboration, and to our families for their unwavering encouragement.

Funding Statement: This work is supported by the National Key Research and Development Program of China (No. 2023YFF0905400) and the National Natural Science Foundation of China (No. U2341229).

Author Contributions: Haitao Wang: Methodology, Software, Conceptualization, Investigation, Validation, Writing—original draft. Yuanzhao Guo: Visualization, Investigation. Xiaotong Han: Data curation. Yuan Tian: Supervision, Project administration, Resources, Writing—review and editing. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The sources of all datasets are cited in the paper, and can be accessed through the links or GitHub repositories provided in their corresponding papers. Other data content can be obtained by contacting the authors. Access links for datasets: DialogRE: https://github.com/nlpdata/dialogre (accessed on 14 January 2025).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Zhang Y, Zhong V, Chen D, Angeli G, Manning CD. Position-aware attention and supervised data improve slot filling. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing; 2017; Copenhagen, Denmark. p. 35–45.
2. Zhou W, Chen M. An improved baseline for sentence-level relation extraction. In: Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers); 2022. p. 161–8.
3. Chen Z, Li Z, Zeng Y, Zhang C, Ma H. GAP: a novel Generative context-Aware Prompt-tuning method for relation extraction. Expert Syst Appl. 2024;248(6):123478. doi:10.1016/j.eswa.2024.123478.
4. Liu Z, Chen X, Wang H, Liu X. Integrating regular expressions into neural networks for relation extraction. Expert Syst Appl. 2024;252(4):124252. doi:10.1016/j.eswa.2024.124252.
5. Wei C, Li J, Wang Z, Wan S, Guo M. Graph convolutional networks embedding textual structure information for relation extraction. Comput Mater Contin. 2024;79(2):3299–314. doi:10.32604/cmc.2024.047811.
6. Yin L, Meng X, Li J, Sun J. Relation extraction for massive news texts. Comput Mater Contin. 2019;60(1):275–85. doi:10.32604/cmc.2019.05556.
7. Han X, Zhao W, Ding N, Liu Z, Sun M. PTR: prompt tuning with rules for text classification. AI Open. 2022;3(10):182–92. doi:10.1016/j.aiopen.2022.11.003.
8. Lee J, Seo S, Choi YS. Semantic relation classification via bidirectional LSTM networks with entity-aware attention using latent entity typing. Symmetry. 2019;11(6):785. doi:10.3390/sym11060785.
9. Ding K, Liu S, Zhang Y, Zhang H, Zhang X, Wu T, et al. A knowledge-enriched and span-based network for joint entity and relation extraction. Comput Mater Contin. 2021;68(1):377–89. doi:10.32604/cmc.2021.016301.
10. Zeng Y, Li Z, Chen Z, Ma H. Aspect-level sentiment analysis based on semantic heterogeneous graph convolutional network. Front Comput Sci. 2023;17(6):176340. doi:10.1007/s11704-022-2256-5.
11. Yang R, Chen Y, Yan J, Qin Y. A graph with adaptive adjacency matrix for relation extraction. Comput Mater Contin. 2024;80(3):4129–47. doi:10.32604/cmc.2024.051675.
12. Choi JD, Chen HY. SemEval 2018 Task 4: character identification on multiparty dialogues. In: Proceedings of the 12th International Workshop on Semantic Evaluation; 2018; New Orleans, LA, USA. p. 57–64.
13. Peng B, Li X, Gao J, Liu J, Wong KF. Deep Dyna-Q: integrating planning for task-completion dialogue policy learning. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers); 2018; Melbourne, VIC, Australia. p. 2182–92.
14. Su SY, Li X, Gao J, Liu J, Chen YN. Discriminative deep Dyna-Q: robust planning for dialogue policy learning. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing; 2018; Brussels, Belgium. p. 3813–23.
15. Tan J, Hu J, Dong S. Incorporating entity-level knowledge in pretrained language model for biomedical dense retrieval. Comput Biol Med. 2023;166(4):107535. doi:10.1016/j.compbiomed.2023.107535.
16. Yu D, Sun K, Cardie C, Yu D. Dialogue-based relation extraction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics; 2020. p. 4927–40.

17. Lee B, Choi YS. Graph based network with contextualized representations of turns in dialogue. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing; 2021; Punta Cana, Dominican Republic. p. 443–55.
18. Long X, Niu S, Li Y. Consistent inference for dialogue relation extraction. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence; 2021; Montreal, QC, Canada. p. 3885–91.
19. Xue F, Sun A, Zhang H, Ni J, Chng ES. An embarrassingly simple model for dialogue relation extraction. In: ICASSP 2022–2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2022; Singapore. p. 6707–11.
20. Fei H, Li J, Wu S, Li C, Ji D, Li F. Global inference with explicit syntactic and discourse structures for dialogue-level relation extraction. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence; 2022; Vienna, Austria. p. 4107–13.
21. Chen H, Hong P, Han W, Majumder N, Poria S. Dialogue relation extraction with document-level heterogeneous graph attention networks. Cognit Comput. 2023;15(2):793–802. doi:10.1007/s12559-023-10110-1.
22. Xue F, Sun A, Zhang H, Chng ES. Gdpnet: refining latent multi-view graph for relation extraction. In: The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21); 2021. p. 14194–202.
23. Wang D, Liu Y. A pilot study of opinion summarization in conversations. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies; 2011; Portland, OR, USA. p. 331–9.
24. Lin PW, Su SY, Chen YN. TREND: trigger-enhanced relation-extraction network for dialogues. In: Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue; 2022; Edinburgh, UK. p. 623–9.
25. Son J, Kim J, Lim J, Lim H. GRASP: guiding model with RelAtional semantics using prompt for dialogue relation extraction. In: Proceedings of the 29th International Conference on Computational Linguistics; 2022; Gyeongju, Republic of Korea. p. 412–23.
26. An H, Zhu Z, Cheng X, Huang Z, Zou Y. Knowledge-enhanced prompt tuning for dialogue-based relation extraction with trigger and label semantic. In: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024); 2024; Torino, Italy. p. 9822–31.
27. Rajpurkar P, Zhang J, Lopyrev K, Liang P. SQuAD: 100,000+ questions for machine comprehension of text. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing; 2016; Austin, TX, USA. p. 2383–92.
28. Bronstein O, Dagan I, Li Q, Ji H, Frank A. Seed-based event trigger labeling: how far can event descriptions get us?. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers); 2015; Beijing, China. p. 372–6.
29. An H, Chen D, Xu W, Zhu Z, Zou Y. TLAG: an informative trigger and label-aware knowledge guided model for dialogue-based relation extraction. In: 2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD); 2023; Rio de Janeiro, Brazil. p. 59–64.
30. Zeng D, Xiao Y, Wang J, Dai Y, Sangaiah AK. Distant supervised relation extraction with cost-sensitive loss. Comput Mater Contin. 2019;60(3):1251–61. doi:10.32604/cmc.2019.06100.
31. Wei J, Wang X, Schuurmans D, Bosma M, Ichter B, Xia F, et al. Chain-of-thought prompting elicits reasoning in large language models. Adv Neural Inf Process Syst. 2022;35:24824–37.
32. Vukovic R, Arps D, van Niekerk C, Ruppik BM, Hc Lin, Heck M, et al. Dialogue ontology relation extraction via constrained chain-of-thought decoding. In: Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue; 2024; Kyoto, Japan. p. 370–84.
33. Kojima T, Gu SS, Reid M, Matsuo Y, Iwasawa Y. Large language models are zero-shot reasoners. Adv Neural Inf Process Syst. 2022;35:22199–213.
34. Xie T, Li Q, Zhang J, Zhang Y, Liu Z, Wang H. Empirical study of zero-shot NER with ChatGPT. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing; 2023; Singapore. p. 7935–56.

35. Devlin J, Chang MW, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT 2019; 2019; Minneapolis, MN, USA. p. 4171–86.
36. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. RoBERTa: a robustly optimized BERT pretraining approach. arXiv:1907.11692. 2019.
37. Chen X, Zhang N, Xie X, Deng S, Yao Y, Tan C, et al. Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In: WWW '22: Proceedings of the ACM Web Conference 2022; 2022; New York, NY, USA. p. 2778–88.
38. Alexis C, Guillaume L. Crosslingual language model pretraining. In: 33rd Conference on Neural Information Processing Systems (NeurIPS 2019); 2019; Vancouver, BC, Canada. p. 7057–67.
39. Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R. ALBERT: a lite BERT for self-supervised learning of language representations. In: Proceeding of ICLR; 2020; Addis Ababa, Ethiopia. p. 3982–92.
40. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is All you Need. In: 31st Conference on Neural Information Processing Systems (NIPS 2017); 2017; Long Beach, CA, USA.
41. Ma X, Zhang Z, Zhao H. Enhanced speaker-aware multi-party multi-turn dialogue comprehension. IEEE/ACM Transact Audio, Speech, Lang Process. 2021;31:2410–23. doi:10.1109/TASLP.2023.3284516.
42. Lee K, Salant S, Kwiatkowski T, Parikh A, Das D, Berant J. Learning recurrent span representations for extractive question answering. arXiv:1611.01436. 2017.
43. Reimers N, Gurevych I. Sentence-BERT: sentence embeddings using siamese BERT-Networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing; 2019; Hong Kong, China. p. 3982–92.
44. Sun W, Yan L, Ma X, Wang S, Ren P, Chen Z, et al. Is ChatGPT good at search? investigating large language models as re-ranking agents. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing; 2023; Singapore. p. 14918–37.
45. Breunig MM, Kriegel HP, Ng RT, Sander J. LOF: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data; 2000; Dallas, TX, USA. p. 93–104.