CLFM: Contrastive Learning and Filter-attention Mechanism for Joint Relation Extraction

Zhiyuan Wang, Chuyuan Wei*, Jinzhe Li, Lei Zhang, Cheng Lv
School of Electrical and Information Engineering,
Beijing University of Civil Engineering and Architecture
Beijing 100044 China

Abstract—**Relation extraction is a fundamental task in natural language processing, which involves extracting structured information from textual data. Despite the success of joint methods in recent years, most of them still have the propagation of cascade errors. Specifically, the error in former step will be accumulated into the final combined triples. Meanwhile, these methods also encounter another challenges related to insufficient interaction between subtasks. To alleviate these issues, this paper proposes a novel joint relation extraction model that integrates a contrastive learning approach and a filter-attention mechanism. The proposed model incorporates a potential relation decoder that utilizes contrastive learning to reduce error propagation and enhance the accuracy of relation classification, particularly in scenarios involving multiple relationships. It also includes a relation-specific sequence tagging decoder that employs a filter-attention mechanism to highlight more informative features, alongside an auxiliary matrix that amalgamates information related to entity pairs. Extensive experiments are conducted on two public datasets and the results demonstrate that this approach outperforms other models with the same structure in recall and F1. Moreover, experiments show that both the contrastive learning strategy and the proposed filter-attention mechanism work well.**

Keywords—Natural language processing; relation extraction; attention mechanism; contrastive learning; multi-task learning

I. Introduction

Relation extraction intend to extract pairs of entity and correlative relations in the form of $< subject, relation, object >$ from the given unstructured texts. The extracted information provides a supplement to many natural language processing(NLP) tasks, such as text summarization [1] knowledge graph construction [2] and question answering [3].

Conventionally, existing methods mainly include pipeline methods and joint methods. Pipeline works [4], [5], [6], [7] traditionally treat the task as two independent subtasks: named entity recognition (NER) and relation extraction (RE). While these approaches are straightforward and adaptable, they overlook the inherent connection between NER and RE, making them prone to error propagation due to the conventional order of subtasks. For this reason, most recent studies focus on joint methods [8], [9], [10]. Current joint models, as evidenced by the work of Cabot et al. in REBEL [11] and Zheng et al. in PRGC [12], have demonstrated remarkable efficiency while achieving outstanding performance. However, most of them first identify entities then find corresponding relationships from all predefined relationships, which faces the problem of relational redundancy and cause unnecessary calculations. Fig. 1 shows the difference between entity first and relation first methods. The entity first method always recognizes possible entity pairs and then matches them against all predefined relationships, which causes redundancy in relationships and introduces unnecessary computation into the model. In contrast, the relation first approach avoids this problem well by first recognizing the relations present in the sentence. Another problem is that some methods only perform simple interactive behaviors between sequence and potential relation representations like concatenating. The operation could carry information unrelated to the task and cannot fully utilize useful mutual information. In addition to this, most of the models suffer from General issue of error propagation. The error in former step will be accumulated in the final triplets.

To alleviate error propagation issue and enhance the interaction between two subtasks, this work makes use of a contrastive learning strategy and proposes a filter-attention mechanism for RE and relation-specific NER(CLFM), respectively. Specifically, the contrastive learning employ the R-drop [13] idea, which has been used in supervised image classification for computer vision tasks. The contrastive learning strategy brings a new constraint to the part of potential relation classification rather than just relying on cross-entropy loss. This can lead to more accurate classification results, especially in scenarios with multiple relationships. The potential relation classification module employs the R-drop idea, which has been used in supervised image classification for computer vision tasks. This strategy brings a new constraint to the part of potential relation classification rather than just relying on cross-entropy loss. This can lead to more accurate classification results, especially in scenarios with multiple relationships, thus the problem of error propagation can be mitigated. For the relation-specific NER task, a novel filter-attention mechanism based on the attention mechanism [14] is proposed. Unlike prior works that simply concatenate or add sentence and relation representations, task interaction in this method is achieved in two ways: Initially, attention scores are computed for sentence and relation representations to signify the token relevance concerning the current relation. Next, low scores indicating weak correlations within the score matrix are removed, followed by the concatenation of particular representations to form the input for the relation-specific Named Entity Recognition (NER) task. Regarding NER, it is perceived as a conventional sequence tagging task for acquiring potential entity pairs. This process strengthens the interaction between

*Corresponding authors

the two subtasks and eliminates inconsequential information across distinct relationships. While executing the relation classification task, simultaneous computation of the auxiliary matrix occurs, which is subsequently utilized to determine the final triplets.

CLFM is mainly composed of three parts: All possible relations in the input sentence are identified in the first part; The second part is the identification of all possible head entities and tail entities under the specific relationships that have been extracted earlier; Finally, the model uses an auxiliary matrix called subject-object alignment step to help select the final triplets from extracted entity pairs and relationships. Since the order of the subtasks, the proposed end-to-end method can also solve the problem of redundant relationships.

The approach has been tested on two widely-used public datasets, namely, NYT [15] and WebNLG [16]. Experimental results indicate the model's performance is on par with state-of-the-art methods on these benchmark datasets. In summary, the paper's key contributions are as follows:

1. A relation first end-to-end framework is introduced by this work, along with the design of three components pertaining to the subtasks. These components effectively tackle issues related to redundant relations and enhance the interaction between subtasks.

2. This work incorporates a contrastive learning strategy into relation classification to introduce a new constraint that enhances classification accuracy and mitigates error propagation.

3. An innovative filter-attention mechanism is presented in this work, with the objective of fostering deeper interaction between the two subtasks by proficiently filtering out irrelevant task-specific information. Extensive experimentation on various public benchmarks demonstrates results that exhibit performance comparable to the baseline, especially in scenarios involving multiple relationships.

This paper is structured as follows: Section II is related work of recent years in relation extraction; Section III are detailed description of the each model part; Section IV are experimental results compared to other baseline methods and ablation experiments; Section V are conclusion and future work.

II. RELATED WORK

The traditional incipient relational triplet extraction methods, such as those proposed by Zelenko et al. [4] and Chan et al. [5], adopted a pipeline framework that divided the entire task into two separate subtasks: entity identification and relation classification. However, these approaches were prone to error propagation problems and ignored the fact that these subtasks are interactive. Therefore, later works began to extract entities and relations jointly using a single model, such as feature-based models [17], [18], [19]. These models rely on various external NLP tools and complicated manual operations, making them heavily dependent on the accuracy of these external tools. Although these models are very representative, they have limitations in terms of scalability and efficiency. In the past few years, joint methods based on neural networks have become a major research focus. This section is presented in three subsections: entity first methods, relation first methods and other methods.

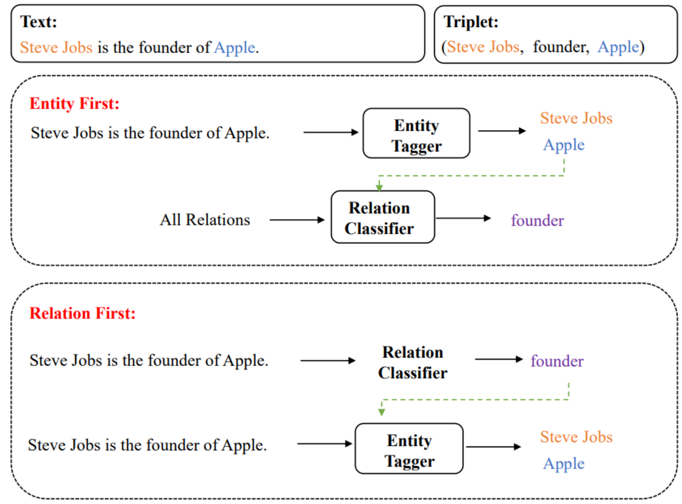

Fig. 1. The difference between entity first and relation first approaches.

A. Entity First

Zheng et al. [20] first proposed a novel joint model based on a tagging scheme, which transformed the relation extraction task into a sequence labeling task and applied Long Short-Term Memory network to learn long-term dependencies. However, the model has no ability to extract overlapping triplets. To address the overlapping issue, Yu et al. [21] proposed a method that extracts head entities in the first step, followed by all correlative tail entities and relations using decoding strategies. In addition, A unified joint extraction annotation framework was designed by Wei et al. [22], capable of achieving single-stage joint extraction while addressing exposure bias and the intricate issue of overlapping. In the work by Wang et al. [23], a one-stage approach was presented to simultaneously extract entities and overlapping relations. This approach effectively narrows the discrepancy between training and inference stages. Specifically, they formulated joint extraction as a token pair linking problem and introduced a novel handshaking tagging scheme that aligns the boundary tokens of entity pairs under each relation type. Shang et al. [24] first generated candidate entities by enumerating sequences of tokens in a sentence, and they converted the extraction task into a linking problem on a head-to-tail bipartite graph that could directly extract all triplets in a single step. Nayak et al. [25] proposed a pointer network-based decoding approach where an entire tuple is generated at every time step and achieved significantly higher F1 scores

B. Relation First

All of the approaches mentioned, regardless of being single-stage or not, suffer from relational redundancy issue [20], [22], [23]. As a result, a new model structure for extracting sequences has emerged. These methods typically involve performing relation classification first, which only preserves related relations and not all redundant relations in the input sentence. However, Yuan et al. [8] proposed a gating mechanism to obtain relation-specific sentence representation, which can be used for sequence tagging tasks and can provide a fine-grained representation. Nevertheless, this mechanism is unable to address the problem of subject-object overlapping. To address this issue, Ma et al. [26] proposed a cascade dual-decoder approach to extract overlapping relational triples. This approach utilizes relevant information of relations and subjects as auxiliary information for subjects and objects recognition, respectively. However, the approach still has poor generalization due to insufficient interaction and a span-based extraction strategy. Zheng et al. [12] decomposed the entire task into three subtasks: relation judgement, entity extraction, and subject-object alignment. They designed a low complexity global correspondence matrix to align the subject and object. Despite achieving success, this approach lacks deep interaction between relation classification and entity recognition.

C. Other Method

Shang et al. [27] proposed a one-step and one-module model that consists of a scoring-based classifier and a relation-specific horns tagging strategy. Zhao et al. [28] tackled the task of relation extraction using heterogeneous graph neural networks. Their approach involves modeling relations and words as nodes on a graph and iteratively fusing the two types of semantic nodes using a message passing mechanism to obtain node representation. This approach leverages the graph structure and takes into account the contextual information of both relations and words. Ning et al. [29] considered the extraction task based on the table-filling method as a target detection task and proposed a single-stage target detection framework, which combined with the auxiliary global relational triplets region detection to ensure the region information can be fully utilized. Ye et al. [30] employed the different strategies for NER and RE by using solid and levitated markers of neighboring spans inside the same sample.

III. Proposed Method

A. Problem Formulation

Given the input sentence $S = \{x_1, x_2, \ldots, x_n\}$ with n tokens, the task goal is to extract all possible relational triplets such as $\{T = (s, r, o) \mid s, o \in E, r \in R\}$, where E and R are entity and relation sets respectively, a triplet T represents a pair of entity and a relation between them contained in sentence S.

As shown in Fig. 2, given an input sentence S, the encoder starts modeling its text semantics. The potential relation decoder, with a contrastive learning strategy, detects all possible relations $r \in R$ based on the text semantics. For each detected relation r, the filter-attention mechanism computes and filters the low weights of each input token to get relation-specific sentence representation as the input of NER. The relation-specific entity decoder extracts the corresponding head and tail entities by using a sequence tagging scheme. Finally, the model obtains the final triplets T with the aid of an auxiliary matrix M which were generated at the same stage as the potential relation prediction.

B. Model Encoder

This approach employs a pre-trained language model called BERT [31] for a fair comparison, which is widely used to encode sentences and capture the semantics of text. The output of the model encoder is $H_{enc} = \{h_1, h_2, \ldots, h_n \mid h_i \in \mathbb{R}^{d\times 1}\}$, where n is the number of tokens, and d is the dimension of the embedding. It can also use other pre-trained language models, such as RoBERTa [32] and so on.

C. Relation Classification

The relation classification component is illustrated in Fig. 2. It is important to note that not all sentences contain all predefined relations. Therefore, CLFM starts by identifying potential relations, which helps to reduce redundant relationships in the current text. To complete the operation, average pooling and a fully connected layer are employed. Given the embedding $\mathbf{h} \in \mathbb{R}^{n\times d}$ of the input sentence, where n is the number of tokens, each element of the relation classification component is obtained as follows:

$$h^{pool} = \varphi(\mathbf{h}) \in \mathbb{R}^{d\times 1}$$
$$P^{pot} = \sigma(W_r h^{pool} + b_r) \tag{1}$$

Where φ denotes the average pooling operation [33] and σ is the sigmoid activation function, $W_r \in \mathbb{R}^{d\times 1}$ is a trainable weight and b_r is a parameter.

Unlike previous works [21], [26], [12], which treat relation classification as a simple binary classification task, this approach incorporates a contrastive learning strategy to add a new constraint to the result of the relation classification task. Inspired by [13], CLFM employs the idea of R-drop to impose a new constraint on the result of the potential relation classification task. Specifically, it computes the Kullback-Leibler divergence of the classification results as a part of the relational loss by running the sentence embedding $\mathbf{h}$ through the sentence classifier twice. Since the classifier contains a dropout operation, two results for the same input may be different, which can increase the robustness of the classifier by continuous training. For classification results, if the probability exceeds a threshold λ_1, model allocates the corresponding relation a tag of 1; otherwise, model assigns a tag of 0. The detailed contrastive learning component and potential relation loss are as follows:

$$L_{cl} = \frac{1}{2}(D_{KL}(P_1^{pot} \parallel P_2^{pot}) + D_{KL}(P_2^{pot} \parallel P_1^{pot})) \tag{2}$$

$$\mathcal{L}_{rc} = -\frac{1}{n_r}\sum_{i=1}^{n_r}(y_i \log P^{pot} + (1-y_i)\log\left(1-P^{pot}\right)) \tag{3}$$

$$L_{rel} = \alpha L_{rc} + \beta L_{cl} \tag{4}$$

Where P_1^{pot} and P_2^{pot} are transformed into a predefined relational representation by P^{pot}, $D_{KL}(\|)$ denotes Kullback-Leibler divergence and n_r is the size of predefined relation set. α and β are weights of each sub-loss. Performance might be better by carefully tuning the weight of each sub-loss. The reason why this work takes the average of the two calculations as the result is that Kullback-Leibler divergence is asymmetric. After adding the new constraint, CLFM can better handle of datasets with more relationships.

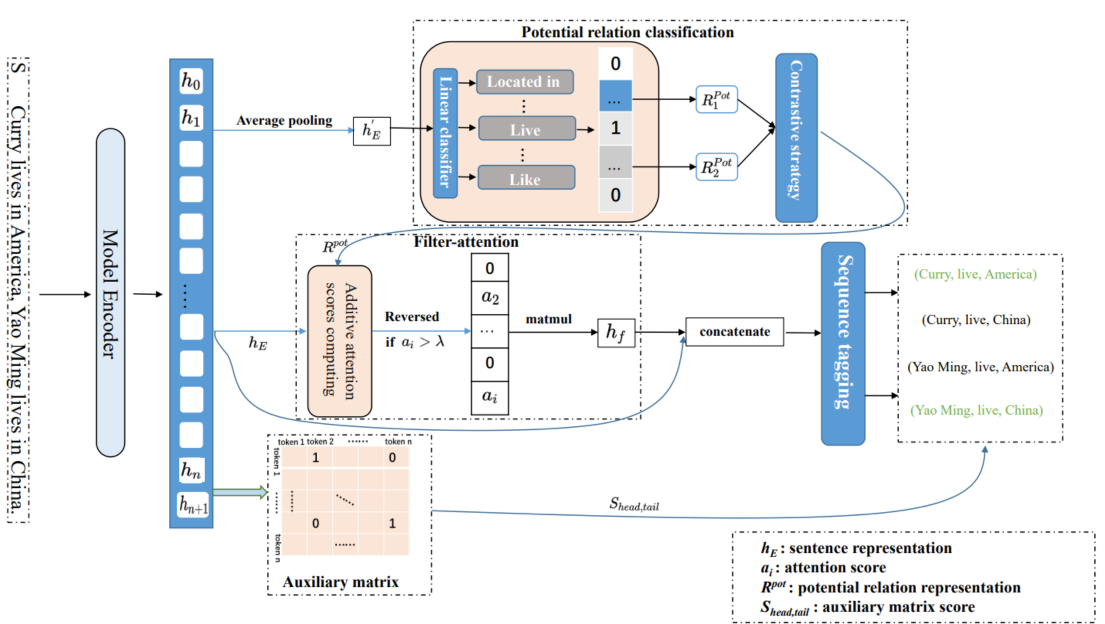

Fig. 2. The overall structure of CLFM. It combines three parts: potential relation classification, sequence labeling and auxiliary matrix.

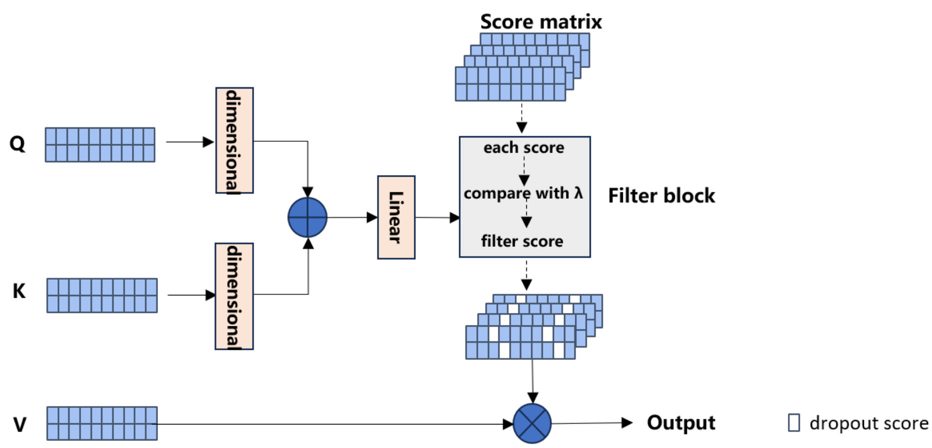

Fig. 3. The process of filter-attention.

D. Filter-attention Mechanism

After obtaining all potential relationships in the current sentence, previous works typically concatenate sentence and relation embeddings or design complex gating mechanism. However, concatenation can lead to the introduction of useless information. Compared to than gating mechanism, attention mechanism is more intuitive and easier to understand and the elements are more closely related. Therefore, this paper designs an attention mechanism with a filtering function based on attention mechanism [14] to retain useful mutual information. Fig. 3 shows the details of filter-attention mechanism. Before starting relation-specific NER, sentence embedding and relation embedding are fed into filter-attention. It utilizes additive attention to capture diverse semantic information across various relationships, as it aligns better with the encoder-decoder structure. After computing attention scores, filter module selects higher scores for retention to obtain a more accurate representation of the input containing information about the current relationship. The filter-attention mechanism component is as follows:

$$\begin{aligned} &S(\mathbf{h}, h_r) = W_v \theta(W_q \mathbf{h} + W_k h_r) \\ &S_{filter}(\mathbf{h}, h_r) = F(Softmax(S(\mathbf{h}, h_r))) \\ &\mu = S_{filter}(\mathbf{h}, h_r) \cdot h_r \end{aligned} \tag{5}$$

Where W_q, $W_k \in \mathbb{R}^{d \times d}$ and $W_v \in \mathbb{R}^{d \times 1}$ are trainable weights, h_r is transformed into a predefined relation representation by P^{pot}. $\theta(\cdot)$ is tanh activation function and $F(\cdot)$ is the filter operation. A threshold λ_{att} is established for the filter mechanism. When scores exceed λ_{att}, the scores in the original matrix are retained; otherwise, they are discarded. After conducting thorough experiments, λ_{att} is assigned a value of 5e-3. The ultimate representation that undergoes the filter-attention mechanism is denoted as $u \in \mathbb{R}^{d \times 1}$.

E. Relation-specific NER

As shown in Fig.2, the model starts the relation-specific entity recognition task after completing the filter-attention operation. CLFM model it as a sequence tagging task because the generalization of span-based extraction methods is poor. To solve common overlapping triplet issues, it perform separate sequence tagging for head entity and tail entity. This strategy can handle issues including EntityPairOverlap (EPO) and SingleEntityOverlap (SEO). The sequence tag set for the head entity is $\{B\text{-}H, I\text{-}H, O\}$, and the sequence tag set for the tail entity is $\{B\text{-}T, I\text{-}T, O\}$. CLFM employ the traditional LSTM-CRF[10] network for sequence tagging, and the detailed formula descriptions are as follows:

$$\begin{aligned} o_{i,j} &= \left[\overrightarrow{\mathrm{LSTM}}\left(h_i; \mu_j\right); \overleftarrow{\mathrm{LSTM}}\left(h_i; \mu_j\right)\right] \\ P_{i,j}^{head} &= \mathrm{CRF}\left(o_{i,j}\right) \end{aligned} \tag{6}$$

Where $(;)$ denotes concatenating operation, h_i is $i\text{-}th$ token representation of sentence S and μ_j is $j\text{-}th$ relation representation after going through filter-attention component. $\mathrm{CRF}(\cdot)$ is Conditional Random Field approach. The head entity formula is only given, tail entity formula is the same as Eq. (6). The loss of whole entity recognition is Eq. (7):

$$\mathcal{L}_{seq} = -\frac{1}{2 \times n \times n_r^{pot}} \sum_{t \in \{head, tail\}} \sum_{j=1}^{n_r^{pot}} \sum_{i=1}^{n} y_{i,j}^t \log P_{i,j}^t \tag{7}$$

Where n_r^{pot} is the size of potential relation set of sentence S.

F. Auxiliary Matrix and Training Strategy

After all the above operations, CLFM gets all possible head and tail entities that correspond to specific relations. However, if the model directly outputs the outcome, there will be a high probability of obtaining inaccurate triplets. To avoid this situation, following [12], the model computes an auxiliary matrix $M \in \mathbb{R}^{n \times n}$ for the given sentence S with n tokens to denote whether a relationship exists between tokens. This is similar to a pruning operation that increases the limits and make the final triplets more accurate. The value of each element in the matrix is computed as follows:

$$P_{i_{head}, j_{tail}} = \sigma\left(W_g\left[h_i^{\text{head}}; h_j^{\text{tail}}\right] + b_g\right) \tag{8}$$

Where h_i^{head}, $h_j^{\text{tail}} \in \mathbb{R}^{d \times 1}$ are the encoded representation of the $i\text{-}th$ token and $j\text{-}th$ token in the input sentence forming a potential pair of head and tail entities. W_g is a trainable weight.

Matrix loss is as follows:

$$\begin{aligned} \mathcal{L}_{\text{matrix}} = -\frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} & \left(y_{i,j} \log P_{i_{\text{head}}, j_{\text{tail}}}\right. \\ & \left. + \left(1 - y_{i,j}\right) \log\left(1 - P_{i_{\text{head}}, j_{\text{tail}}}\right)\right) \end{aligned} \tag{9}$$

The total loss is the sum of these three parts:

$$\mathcal{L}_{totalloss} = \gamma_1 L_{rel} + \gamma_2 L_{seq} + \gamma_3 L_{matrix} \tag{10}$$

where $\gamma_1, \gamma_2, \gamma_3$ are adjustable loss weights.

IV. Experiment

A. Datasets and Evaluation Metric

For fair and comprehensive comparison, this work follow [22], [9], and [21] to evaluate CLFM on two widely used public datasets: NYT and WebNLG. The NYT dataset is generated by aligning the relations in Freebase with the New York Times (NYT) corpus and is widely used for remotely supervised relational extraction tasks. It contains 24 relation types. The WebNLG dataset, originally employed for natural language generation tasks, includes 246 relation types. It is worth noting that both datasets have another version: NYT* and WebNLG*, which annotate the last word of entities. Table I shows the statistics for the above datasets.

Two evaluation metrics are employed for expensive experimental studies: Partial Match for NYT* and WebNLG*, where an extracted triple (s, r, o) is regarded as correct only if its relation and the last word of the head entity name and the tail entity name are correct; Exact Match for NYT and WebNLG, where a predicted triple (s, r, o) is regarded as correct only if its relation and the full names of its head and tail entities are all correct.

TABLE I. The Statistics of Datasets

Datasets	Train	Valid	Test	Relations
NYT	56196	5000	5000	24
WebNLG	5019	500	703	171
NYT*	56195	4999	5000	24
WebNLG*	5019	500	703	216

B. Implementation Details

As presented in Fig. 2, CLFM encoder employs the PyToREh version of $BERT_{base}\,(cased)$ English. To ensure equitable comparison, the input sentence length is standardized to a fixed size of 100, and the Adam optimizer [34] is employed with a batch size of 32/6 for the NYT/WebNLG datasets. The learning rate for the BERT encoder is set to 5e-5, while the decoder learning rate is set to 1e-3 to achieve fast convergence. Moreover, the utilization of weight decay [35] at a rate of 0.01 is incorporated. The potential relation decoder threshold and filter-attention threshold are set as 0.5 and 5e-3, respectively.

The experiments are conducted on a server equipped with Intel(R) Xeon(R) Silver 4215 CPU @ 2.50GHz and an NVIDIA Tesla V100 GPU.

C. Baseline Methods

A selection of nine baseline methods has been made for the purpose of comparison. This assortment comprises representative models as well as models featuring analogous structures. CLFM is compared with the following strong baseline models on the NYT and WebNLG datasets. The top six models are representative methods, while the last three have similar structures to CLFM: (1) CasRel[22] (2) TPLinker[23] (3)WDec[25](4) CGT[36] (5) StereoRel[37] (6) RIFRE[28] (7)PRGC[12] (8) RSAN[8] (9) Cascade dual-decoder[26].

TABLE II. PERFORMANCE OF CLFM AND EIGHT COMPARED BASELINES ON NYT AND WEBNLG

Model	NYT			WebNLG			NYT*			WebNLG*		
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
CasRel	-	-	-	-	-	-	84.2	83.0	83.6	86.9	80.6	83.7
TPLinker	91.4	92.6	92.0	88.9	84.5	86.7	91.3	92.5	91.9	91.8	92.0	91.9
WDec	88.1	76.1	81.7	88.6	51.3	65.0	94.5	76.2	84.4	-	-	-
CGT	-	-	-	-	-	-	94.7	84.2	89.1	92.9	75.6	83.4
StereoRel	92.0	92.3	92.2	-	-	-	92.0	92.3	92.2	91.6	92.6	92.1
RIFRE	-	-	-	-	-	-	93.6	90.5	92.0	93.3	92.0	92.6
PRGC	93.5	91.9	92.7	89.9	87.2	88.5	93.3	91.9	92.6	94.0	92.1	93.0
RSAN	85.7	83.6	84.6	80.5	83.8	82.1	-	-	-	-	-	-
Cascade dual-decoder	89.9	91.4	90.6	88.0	88.9	88.4	90.2	90.9	90.5	90.3	91.5	90.9
CLFM	93.3	**92.4**	**92.8**	**90.3**	87.9	**89.1**	93.0	92.3	**92.7**	93.9	**92.6**	**93.3**

D. Experimental Results

This section presents experimental results and compares them with other baseline models. It also conduct an in-depth analysis of the results to gain a better understanding of the performance of CLFM in relation to the other methods. Through a comprehensive analysis of the outcomes, valuable insights into the workings of CLFM can be acquired.

1) Overall Results: Table II presents an overall comparison of CLFM with other baselines. CLFM outperforms all of the baselines in terms of F1 scores, and for most cases, precision and recall are also superior. Notably, CLFM exhibits better robustness and generalization on the WebNlG dataset, where a wide range of relations are involved. This success can be attributed to the incorporation of the introduced contrastive learning strategy, which significantly enhances the accuracy of the relation classification decoder. Although RSAN[8] also employs attention calculations through a gate mechanism, CLFM achieves superior performance (with at least an 8% improvement over RSAN) due to its simplicity and commonality. Additionally, the model achieve a 0.6% improvement on the WebNLG dataset compared to PRGC. Regarding the NYT dataset, although CLFM achieves similar performance as PRGC, it is believe that the limited number of relations in the dataset and the already high-quality results may have contributed to the lack of significant improvement.

2) Result Analysis on Different Sentence Types : Following previous works [26], we conduct extensive experiments to verify that our method makes effective in the scenario of overlapping triples on NYT and WebNLG datasets. Table III shows the detailed results on the three overlapping patterns, where Normal is the easiest pattern while EPO and SEO are more difficult to be handled. The experimental findings reveal a consistent and superior performance exhibited by our proposed model across all three overlapping patterns. Noteworthy is the model's exceptional efficacy in handling intricate patterns such as EPO and SEO, where it consistently outperforms the established baseline, PRGC. This substantiates the robust capabilities of our model in effectively addressing the intricate challenges posed by overlapping triplets.

Furthermore, an extensive examination was conducted to extract triples from sentences featuring varying numbers of triplets. The sentences were categorized into five subclasses, each encompassing texts with 1, 2, 3, 4, or $\geq$ 5 triples. The comparative results of the five methods across the different triple categories are depicted in Fig. 4. The figure illustrates that our model consistently achieves the highest F1 scores across most cases in the two datasets, exhibiting remarkable stability as the number of triples increases. Particularly noteworthy is the superior performance of our model in comparison to the leading baseline, PRGC, in the most challenging class ($\geq$ 5) on NYT and WebNLG. This suggests that our model demonstrates enhanced resilience in handling intricate scenarios involving a substantial number of triples.Moreover, our model outperforms others in nearly every subset, irrespective of the number of triples. In summary, these additional experiments substantiate the advantageous features of our model, particularly in complex scenarios, highlighting its robustness and superior performance over existing methods.

TABLE III. F1-SCORE OF SENTENCES WITH DIFFERENT OVERLAPPING TRIPLETS ON NYT AND WEBNLG

Model	NYT			WebNLG		
	Normal	SEO	EPO	Normal	SEO	EPO
WDec	80.3	81.4	86.7	75.5	63.3	67.0
CasRel	84.2	83.0	83.6	86.9	80.6	83.7
PRGC	88.6	93.6	94.1	86.8	89.0	89.8
Cascade dual-decoder	88.2	92.8	92.9	86.2	88.9	88.5
CLFM	**90.9**	**94.4**	**94.7**	**87.6**	**89.6**	**94.1**

3) Explore for Filter Threads: The performance of the filter-attention mechanism is highly influenced by the threshold λ_{att}. To study the impact of threshold changes and find the appropriate value, A series of experiments are carried out. Table IV shows the results of these experiments. Variations were introduced to the threshold within a predefined range, revealing that further enhancements in the experimental outcomes could be achieved by implementing additional threshold adjustments. Based on the current results, A threshold value of 5e-3 was chosen for the filter-attention mechanism. Nevertheless, it is important to highlight that the optimal threshold might exhibit variability contingent on the specific dataset or task. As a consequence, undertaking additional experiments and meticulous refinement of the threshold could be imperative

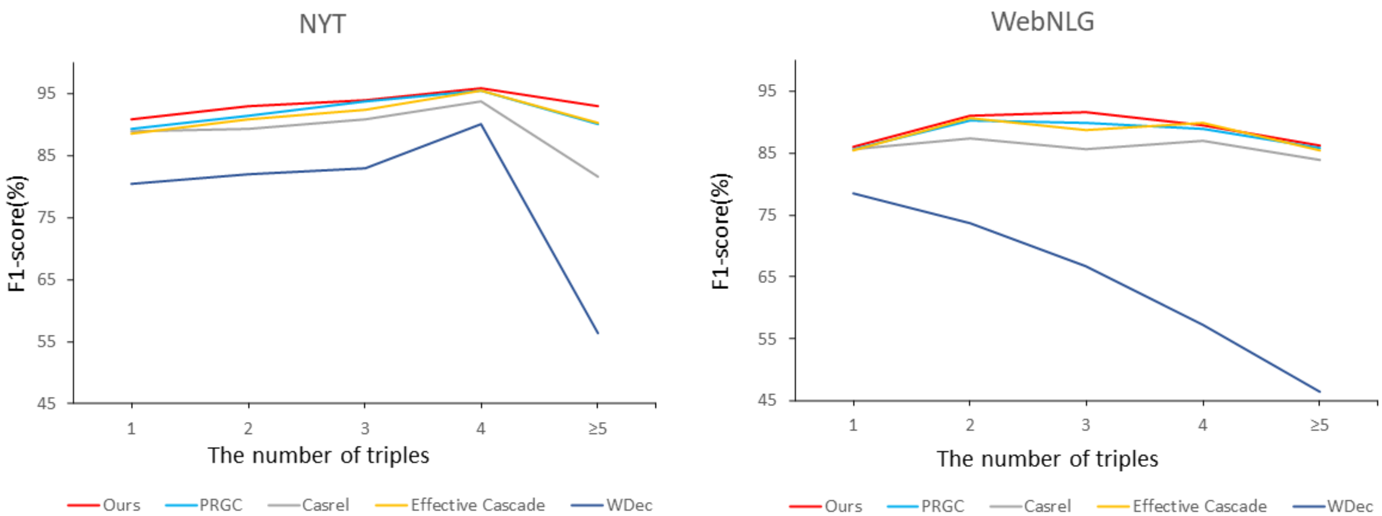

Fig. 4. Results of different sentence types on NYT and WebNLG.

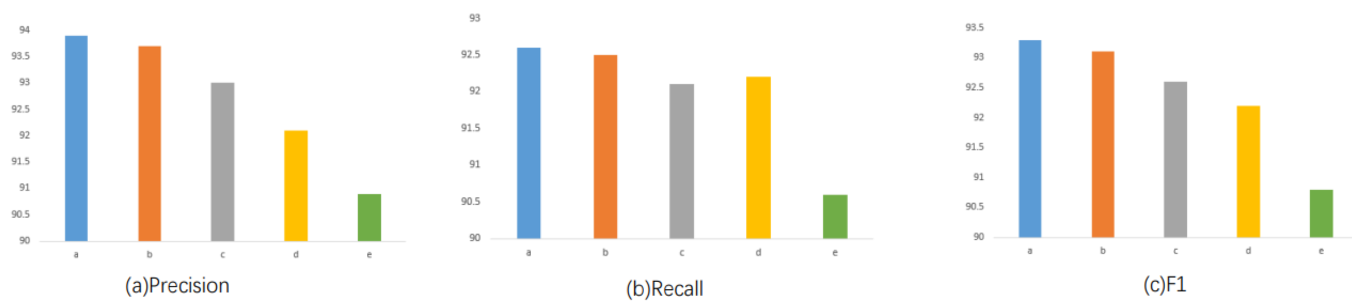

Fig. 5. The results of different combination approaches.

for attaining superior results.

TABLE IV. THE EXPERIMENTAL RESULTS OF DIFFERENT THREADS ON WEBNLG*

Thread	F1(WebNLG*)
1e-3	92.8
3e-3	92.4
4e-3	92.0
5e-3	**93.3**
7e-3	92.0

4) Ways of Filter-attention: Furthermore, diverse methodologies for amalgamating the representations acquired via the filter-attention mechanism with the sentence embeddings are explored. To facilitate discourse, the nomenclature adopted encompasses the relation embeddings procured via the filter-attention mechanism, denoted as filter-relation, the sentence embeddings acquired through the filter-attention mechanism, termed as filter-sequence, and the sentence embeddings generated by the BERT encoder, identified as sentence-output. Concatenation function is represented by the $(\cdot,\cdot)$ notation. As illustrated in Fig. 5, five distinct strategies for amalgamating the embeddings are subjected to experimentation on the WebNLG* dataset, namely, (a) sequence-output, filter-relation, (b) filter-sequence, (c) sequence-output, filter-relation + filter-sequence, (d) sequence-output, filter-sequence, (e) filter-sequence, filter-relation. Precision, recall, and F1 score are used to evaluate the performance of each approach. The experiment results demonstrate that the first Combination method, (sequence-output, filter-relation), outperforms the other four combinations and is currently the best approach. Additionally, this validation substantiates the rationale behind the inception of the filter-attention mechanism.

5) Ablation Study: This section examines the contributions of different modules components in CLFM, using the best performing model on the WebNLG* dataset. Initially, the contrastive learning component is removed. Subsequently, an exploration is conducted into the influence of the filter-attention mechanism. This entails retaining the attention mechanism while discarding the filter function, as well as removing both components. Table V shows the results. It can observe that contrastive strategy and filter-attention mechanism can improve the performance of the model. The results demonstrate that contrastive learning can enhance the accuracy of the relation decoder when dealing with multiple relations, while filter-attention can improve the quality of the interactive representations between two subtasks.

TABLE V. ABLATION STUDY RESULTS ON WEBNLG* TEST SET

Method	Pre.	Rec.	F1
CLFM	**93.9**	**92.6**	**93.3**
-contrastive learning	93.5	92.3	92.9
-filter	93.4	92.2	92.8
-filter-attention	93.4	92.0	92.7

V. CONCLUSION

In this paper, a novel approach is presented for relational triplet extraction, emphasizing a relation-first perspective. This work incorporates a contrastive strategy and a filter-attention mechanism to effectively address the challenges posed by redundant relations and the accurate classification of relations within a diverse range. The proposed model also enhances the synergy between subtasks and effectively filters out extraneous information during sequence tagging. Empirical evaluations conducted on publicly available datasets showcase the remarkable performance of CLFM. In the subsequent research endeavors, the exploration of more intricate filter-attention mechanism is on the horizon to elevate the overall quality of representations.

REFERENCES

[1] V. Gupta and G. S. Lehal, "A survey of text summarization extractive techniques," *Journal of emerging technologies in web intelligence*, vol. 2, no. 3, pp. 258–268, 2010.

[2] S. Riedel, L. Yao, A. McCallum, and B. M. Marlin, "Relation extraction with matrix factorization and universal schemas," *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 74–84, 2013.

[3] D. Diefenbach, V. Lopez, K. Singh, and P. Maret, "Core techniques of question answering systems over knowledge bases: a survey," *Knowledge and Information systems*, vol. 55, pp. 529–569, 2018.

[4] D. Zelenko, C. Aone, and A. Richardella, "Kernel methods for relation extraction," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1083–1106, 2003.

[5] Y. S. Chan and D. Roth, "Exploiting syntactico-semantic structures for relation extraction," *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pp. 551–560, 2011.

[6] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*, pp. 2335–2344, 2014.

[7] Z. Zhong and D. Chen, "A frustratingly easy approach for entity and relation extraction," *arXiv preprint arXiv:2010.12812*, 2020.

[8] Y. Yuan, X. Zhou, S. Pan, Q. Zhu, and L. Guo, "A relation-specific attention network for joint entity and relation extraction," *Twenty-Ninth International Joint Conference on Artificial Intelligence and Seventeenth Pacific Rim International Conference on Artificial Intelligence IJCAI-PRICAI-20*, 2020.

[9] X. Zeng, D. Zeng, S. He, L. Kang, and J. Zhao, "Extracting relational facts by an end-to-end neural model with copy mechanism," *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018.

[10] R. Panchendrarajan and A. Amaresan, "Bidirectional lstm-crf for named entity recognition," *The 32nd Pacific Asia Conference on Language, Information and Computation (PACLIC 32)*, 2019.

[11] P.-L. H. Cabot and R. Navigli, "Rebel: Relation extraction by end-to-end language generation," *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 2370–2381, 2021.

[12] H. Zheng, R. Wen, X. Chen, Y. Yang, Y. Zhang, Z. Zhang, N. Zhang, B. Qin, M. Xu, and Y. Zheng, "Prgc: Potential relation and global correspondence based joint relational triple extraction," *arXiv preprint arXiv:2106.09895*, 2021.

[13] L. Wu, J. Li, Y. Wang, Q. Meng, T. Qin, W. Chen, M. Zhang, T.-Y. Liu *et al.*, "R-drop: Regularized dropout for neural networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 10 890–10 905, 2021.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv*, 2017.

[15] S. Riedel, L. Yao, and A. K. Mccallum, "Modeling relations and their mentions without labeled text," *Springer-Verlag*, 2010.

[16] C. Gardent, A. Shimorina, S. Narayan, and L. Perez-Beltrachini, "Creating training corpora for nlg micro-planning," *Meeting of the Association for Computational Linguistics*, 2017.

[17] X. Yu and W. Lam, "Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach," *Coling 2010: Posters*, pp. 1399–1407, 2010.

[18] M. Miwa and Y. Sasaki, "Modeling joint entity and relation extraction with table representation," *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1858–1869, 2014.

[19] X. Ren, Z. Wu, W. He, M. Qu, C. R. Voss, H. Ji, T. F. Abdelzaher, and J. Han, "Cotype: Joint extraction of typed entities and relations with knowledge bases," *Proceedings of the 26th international conference on world wide web*, pp. 1015–1024, 2017.

[20] S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou, and B. Xu, "Joint extraction of entities and relations based on a novel tagging scheme," *arXiv preprint arXiv:1706.05075*, 2017.

[21] B. Yu, Z. Zhang, X. Shu, Y. Wang, T. Liu, B. Wang, and S. Li, "Joint extraction of entities and relations based on a novel decomposition strategy," *arXiv preprint arXiv:1909.04273*, 2019.

[22] Z. Wei, J. Su, Y. Wang, Y. Tian, and Y. Chang, "A novel cascade binary tagging framework for relational triple extraction," *arXiv preprint arXiv:1909.03227*, 2019.

[23] Y. Wang, B. Yu, Y. Zhang, T. Liu, H. Zhu, and L. Sun, "Tplinker: Single-stage joint extraction of entities and relations through token pair linking," *Proceedings of the 28th International Conference on Computational Linguistics*, 2020.

[24] Y. Shang, H. Huang, X. Sun, W. Wei, and X. Mao, "Relational triple extraction: One step is enough," *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pp. 4360–4366, 2022. [Online]. Available: https://doi.org/10.24963/ijcai.2022/605

[25] T. Nayak and H. T. Ng, "Effective modeling of encoder-decoder architecture for joint entity and relation extraction," in *AAAI*, 2020, pp. 8528–8535.

[26] L. Ma, H. Ren, and X. Zhang, "Effective cascade dual-decoder model for joint entity and relation extraction," *CoRR*, vol. abs/2106.14163, 2021. [Online]. Available: https://arxiv.org/abs/2106.14163

[27] Y. Shang, H. Huang, and X. Mao, "Onerel: Joint entity and relation extraction with one module in one step," *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 11 285–11 293, 2022. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/21379

[28] K. Zhao, H. Xu, Y. Cheng, X. Li, and K. Gao, "Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction," *Knowledge-Based Systems*, vol. 219, p. 106888, 2021.

[29] J. Ning, Z. Yang, Y. Sun, Z. Wang, and H. Lin, "Od-rte: A one-stage object detection framework for relational triple extraction," *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11 120–11 135, 2023.

[30] D. Ye, Y. Lin, P. Li, and M. Sun, "Packed levitated marker for entity and relation extraction," *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 4904–4917, 2022. [Online]. Available: https://doi.org/10.18653/v1/2022.acl-long.337

[31] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019. [Online]. Available: https://doi.org/10.18653/v1/n19-1423

[32] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: http://arxiv.org/abs/1907.11692

[33] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013.

[34] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2014.

[35] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," *CoRR*, vol. abs/1711.05101, 2017. [Online]. Available: http://arxiv.org/abs/1711.05101

[36] H. Ye, N. Zhang, S. Deng, M. Chen, C. Tan, F. Huang, and H. Chen, "Contrastive triple extraction with generative transformer," *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 14 257–14 265, 2021. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17677

[37] X. Tian, L. Jing, L. He, and F. Liu, "Stereorel: Relational triple extraction from a stereoscopic perspective," *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4851–4861, 2021.