applied sciences

MDPI

Article

A Keyword Detection and Context Filtering Method for Document Level Relation Extraction

Hailan Kuang, Haoran Chen *, Xiaolin Ma * and Xinhua Liu

Hubei Key Laboratory of Broadband Wireless Communication and Sensor Networks, School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China; kuanghailan@whut.edu.cn (H.K.); liuxinhua@whut.edu.cn (X.L.)

* Correspondence: harry_chen@whut.edu.cn (H.C.); maxiaolin0615@whut.edu.cn (X.M.)

Abstract: Relation extraction (RE) is the core link of downstream tasks, such as information retrieval, question answering systems, and knowledge graphs. Most of the current mainstream RE technologies focus on the sentence-level corpus, which has great limitations in practical applications. Moreover, the previously proposed models based on graph neural networks or transformers try to obtain context features from the global text, ignoring the importance of local features. In practice, the relation between entity pairs can usually be inferred just through a few keywords. This paper proposes a keyword detection and context filtering method based on the Self-Attention mechanism for document-level RE. In addition, a Self-Attention Memory (SAM) module in ConvLSTM is introduced to process the document context and capture keyword features. By searching for word embeddings with high cross-attention of entity pairs, we update and record critical local features to enhance the performance of the final classification model. The experimental results on three benchmark datasets (DocRED, CDR, and GBA) show that our model achieves advanced performance within open and specialized domain relationship extraction tasks, with up to 0.87% F1 value improvement compared to the state-of-the-art methods. We have also designed experiments to demonstrate that our model can achieve superior results by its stronger contextual filtering capability compared to other methods.

Keywords: relation extraction; keyword detection; Self-Attention; ConvLSTM; context extraction; transformer

Citation: Kuang, H.; Chen, H.; Ma, X.; Liu, X. A Keyword Detection and Context Filtering Method for Document Level Relation Extraction. *Appl. Sci.* **2022**, *12*, 1599. https://doi.org/10.3390/app12031599

Academic Editor: Valentino Santucci

Received: 14 January 2022
Accepted: 30 January 2022
Published: 2 February 2022

1. Introduction

RE is a crucial subtask in the field of information extraction. It can be divided into sentence-level extraction and document-level extraction according to the length of the input text. Previous work [1,2] mainly focused on predicting the relation between entities in the same sentence and could not handle cross-sentence extraction. The authors of [3] found that as many as 40.7% of the relational facts in the long corpus can only be extracted by combining the semantics of multiple sentences based on the statistics of the Wikipedia document corpus. Document-level corpus usually has a larger input length and a more complex structure, which poses great challenges to the information extraction capabilities of existing sentence-level models. Recently, many studies [3,4] have extended sentence-level RE to the document-level.

Many studies have adopted graph neural networks and used structure-dependent, heuristic, or structured attention mechanisms for relational reasoning [5–9]. The document graph they constructed can connect entities at a longer distance, which makes up for the shortcomings of the RNN-based encoder. With the introduction of the transformer model [10], studies have found that this model structure can implicitly capture long-distance dependencies [11,12]. Therefore, many studies have abandoned the graph neural network and used the pre-trained language model BERT [13] for relational reasoning and achieved good results.

However, previous research focuses on integrating global contextual information to enhance entity representation while ignoring the impact of genuinely critical local

information on relation classification. As shown in Figure 1, we take the corpus mentioned in [14] as an example. When classifying the relation between "John Stanistreet" and "Bendigo" in the text, we mark their cross-attention in the pre-training model on the corresponding words. The correct classification relation of this entity pair should be *place of birth* and *place of death*. However, by observing the visualization results, we found that the pre-training model focuses on some entity nouns and punctuation marks that are unrelated to the classification [11]. The "born in" and "died in", which are really helpful for the relation classification, were given little attention by the model. In addition, processing too much global context feature information may dilute the feature information provided by keywords, which will interfere and confuse the final classification task.

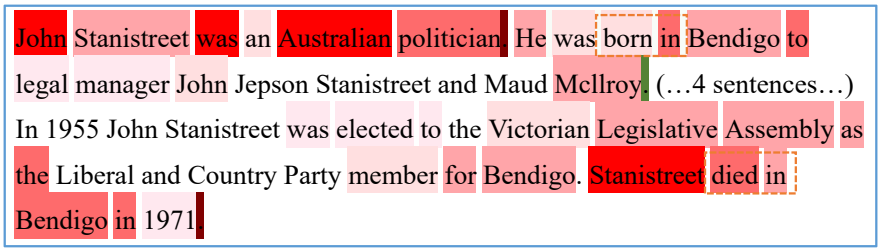

Figure 1. Pre-trained model attention visualization results. The darker color of the marker represents the higher the model's attention to the word.

Based on the above observations, this paper proposes a keyword feature capture method based on the Self-Attention mechanism and uses the SAM module for recording context features in ConvLSTM [15]. By searching for the keyword embeddings with high cross-attention of entities, the local feature information that is helpful for relation classification is recorded to enhance the performance of downstream classification tasks. The contributions of this article are as follows:

- We present a BERT-LSTM-based model in the field of document RE that creatively uses the attention of subject-object entities to match the word embeddings that determine their relation;
- An improved ConvLSTM model is introduced to make it suitable for document sequence extraction. We build the model's attention relation from scratch and enable it to extract useful contextual keyword features;
- Compared with state-of-the-art models, our model has achieved advanced performance on the three benchmark datasets, proving the effectiveness of using keywords for RE.

The rest of the article is organized as follows: Section 2 will describe the previous related work and the foundation of our work. Section 3 will explain how we decompose the RE task and present the overall structure of our method and the processing flow of each part. In Section 4, we report the experimental performance of our model and give a detailed analysis of the experimental results to demonstrate the effectiveness of the method. Section 5 summarizes the work with the limitations and future directions.

2. Related Work

Graph-based models and transformer-based models are the most common approaches for document-level RE. Graph-based methods are now widely adopted in RE because of their intuitiveness and effectiveness in relational reasoning. To cope with the challenge brought by document RE, the authors of [6] proposed an edge-oriented document-level RE graph neural model (EoG). The authors of [16] proposed a graphically enhanced dual attention network (GEDA) for attentional supervision from additional evidence to construct complex interactions between sentences and potential relational instances. The authors of [17] proposed a Dual-tier Heterogeneous Graph (DHG) to model the structural information of documents and perform relational reasoning across sentences. The authors of [18] proposed an encoder-classifier

reconstructor model (HeterGSAN) that reconstructs the paths and puts more attention on related entity pairs instead of negative samples. In general, graph-based methods are popular in current applications and can overcome the problems caused by long-distance dependencies. However, the type of nodes and edges used in the graph-based approach requires delicate manual design, which is not considered elegant, and it is a common problem of the graph-based method on non-graph data.

Many researchers have also proposed the transformer-based approach. The transformer's multi-headed attention component can divide semantic features into multiple subspaces, allowing the model to learn richer contextual features from the text. The authors of [19] observed an unbalanced relation distribution on the DocRED dataset. They proposed a two-step RE mode, separating relation recognition and classification, which can avoid the influence of a large number of negative samples in the dataset. The authors of [20] proposed a hierarchical inference network reasoning network (HIN) for document-level reasoning. It uses a hierarchical reasoning method to aggregate the reasoning information at the entity level, sentence level, and document level. The authors of [21] proposed CorefBERT, which aims to capture co-referential information in the text. They incorporated entity-pair matching, mention-entity matching, and mention-mention matching into the pre-trained model to enhance the representation of the relational extraction model. The authors of [14] proposed a model based on entity-enhanced embedding and adaptive threshold loss function. They used the attention matrix output by the pre-training model BERT to calculate the cross-attention between entities and introduced a threshold class to calculate the loss of positive and negative samples, respectively, which allowed their model to achieve SOTA results on multiple datasets. We can summarize the predecessors' work into two aspects:

- How to construct potentially useful information for determining the existence of relations from a large amount of textual data;
- How to overcome the uneven distribution of relations in the dataset and the negative impact of a large number of negative samples on the model.

In addition, there have been research attempts to extract relations based on keywords in the field of professional documents. The authors of [22] used a domain keyword collection mechanism to guide the model to focus on the semantic interaction between biological entities linked by keywords by manipulating the attention mask, which improved the F1 value by 5.6% on the Biocreative V Chemical Disease Relation (CDR) dataset. The authors of [23] used the BiLSTM neural network and attention mechanism to extract time and sentence-level important information. They selected each sentence's critical feature layer under the pooling layer's action, which has also achieved advanced results in the field of patent document RE. In general, they constructed keyword-attentive models and made them more sensitive to specific words in the specialized domains. However, this approach cannot be generalized to open domains because it is not possible to enumerate all keywords in open domains.

The authors of [24] used a SALSTM model that can mine and capture remote dependencies in the spatiotemporal sequence prediction task in the field of computer vision and achieved the most advanced results on multiple tasks such as precipitation prediction. Similar to precipitation prediction, the RE task is also a sequence feature extraction and prediction task. Inspired by this, we propose a SAM-LSTM model that can be used to process document sequences, integrates a self-attention memory module and trains a new attention layer from scratch. See the next section for specific methods.

3. Methodology

The structure and processing flow of the method proposed in this paper are shown in Figure 2. The word embedding output by the pre-training model is the original input of the SAM-LSTM model, and the Self-Attention mechanism lets the model learn to find the critical feature information. The weighted feature after attention screening is combined with entity mentions and input to the fully connected layer for the final classification.

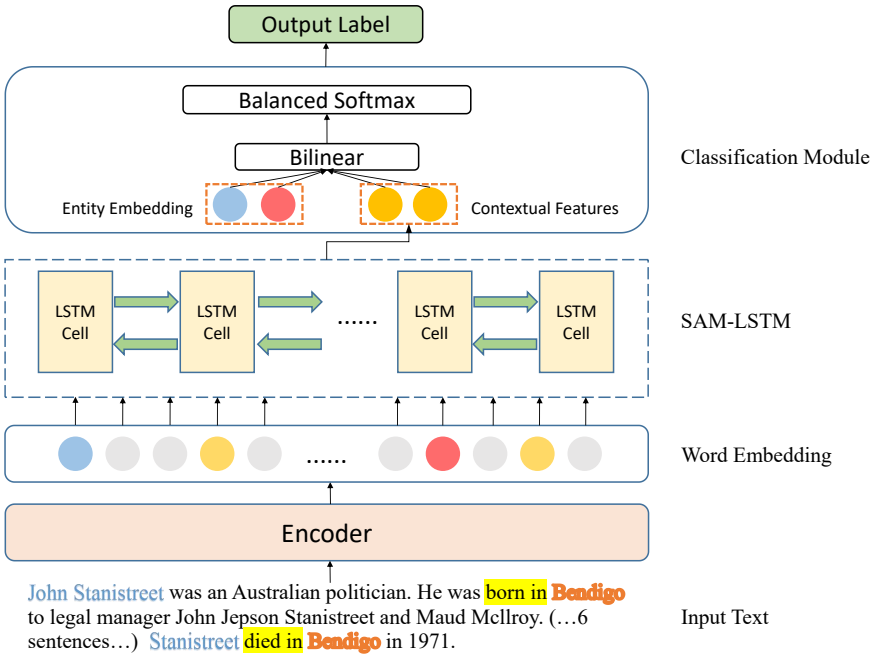

Figure 2. The architecture of our method.

3.1. Encoder

We model the document-level RE task: with a document d, $\{e_i\}_{i=n}^{n}$ is the set of all entities it contained. The aim is to find out the existing relationship from the pre-specified relationship set $R \cup \{NA\}$(NA stands for no relation) for every pair of entity pairs (e_s, e_o) in the set. An entity e_i can be mentioned multiple times in the document as $\{m_k^i\}_{k=1}^{e_i}$. If the entity pair (e_s, e_o) contains relation, then this relation is expressed through one of their mentions. The RE model needs to classify the relation between all entities in document d. In order to further improve the task performance, we adopt the BERT pre-training model and integrate the following technologies to build the entire model. We can think of a document as a sequence of words $d = [x_t]_{t=1}^{l}$. We add the "*" symbol before and after the entity mention to mark the position of the entity. Then, we enter the document into the pre-training model BERT to obtain the word embedding sequence $\boldsymbol{H}$ of the entire document:

$$\boldsymbol{H} = [h_1, h_2, \ldots, h_l] = BERT(x_1, x_2, \cdots, x_l) \tag{1}$$

For some documents whose length exceeds 512, we utilize dynamic windows to encode the entire document, and average the embedding of overlapping marks in different windows to get its final representation. Multiple mentions of an entity may produce different word embeddings, which is not conducive to further classification processing. As a result, logsumexp pooling [25] is used in this paper. The embedding of the entity is obtained using the following formula for multiple mentions.

$$h_{e_i} = \log \sum_{k=1}^{n_{e_i}} \exp\left(h_{m_k^i}\right) \tag{2}$$

This pooling operation is similar to max pooling, but it can better accumulate the mentioned information, and it also shows better performance than average pooling in the experiment.

3.2. Context Feature Extraction

Since the data volume remains large after one document word embedding, this paper performs contextual feature processing based on the ConvLSTM model [15]. This model replaces the input-to-state, state-to-state fully connected operations of ordinary LSTM with convolutional operations, which can establish temporal relations as LSTM and carve local spatial features as CNN. It also enhances feature extraction while reducing the amount of operations. As shown in Figure 3, h_i is the vector in the word embedding sequence, $\boldsymbol{C}_i$ is the state corresponding to cell at step i, $\boldsymbol{H}_i$ is the current input feature, $\boldsymbol{M}_i$ is the sequence information accumulated by cell at step i.

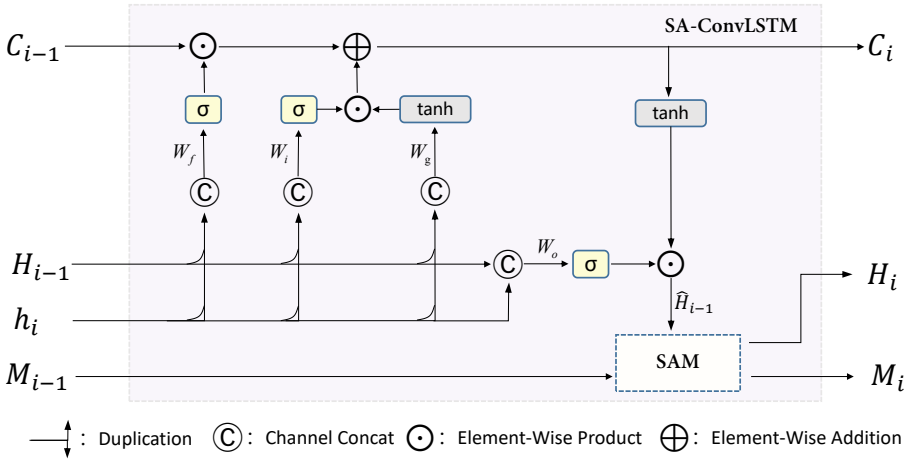

Figure 3. The architecture of SAM-LSTM model. The SAM-LSTM is built by embedding the SAM module into a standard ConvLSTM.

This model sequentially processes the word embedding sequence h_i, controls the cell state $\boldsymbol{C}_i$ by the operations of forgetting gate, input gate and output gate. Meanwhile, the SAM module is used to calculate the attention values with entity embeddings h_{e_s} and h_{e_o} for the memory $\boldsymbol{M}_{i-1}$ and hidden state $\widehat{\boldsymbol{H}}_{i-1}$. Using this attention value to weight the current input features and the features stored in the memory module, we finally output the $\widehat{\boldsymbol{H}}_i$ and update the $\boldsymbol{M}_i$.

3.3. Self-Attention Memory

It has been shown previously that obtaining the attentional relations from the pre-trained model may not yield good results for downstream relational classification tasks. We train a new attentional layer from scratch compared to the previous work [14] which directly uses the multiheaded attention of BERT to process the full-text context. In this paper, a novel SAM module is proposed for capturing key features of full-text, while a memory unit $\boldsymbol{M}$ is used to store contextual sequence information with a global sensory field.

The main idea of the module processing is to use entity pairs to match the word embeddings of interest in the input sequence. Following ConvLSTM processing, the attention matrix of the entity pairs and features is generated, and the parts with greater attention are weighted and updated to be stored in the memory unit. This ensures that the model is able to store the key features of the entity's dependency on the relation even after traversing the entire sequence of contexts.

The detailed structure of the SAM module is shown in Figure 4, where $\widehat{\boldsymbol{H}}_{i-1}$ is the current hidden feature being processed by ConvLSTM, and $\boldsymbol{M}_{i-1}$ is the memory accumulated since the previous step. $\boldsymbol{H}_e$ is the entity pair matrix, which is connected by the subject-object entity embedding h_{e_s} and h_{e_o}. The module can be divided into three parts: self-attentive feature aggregation, memory update, and output.

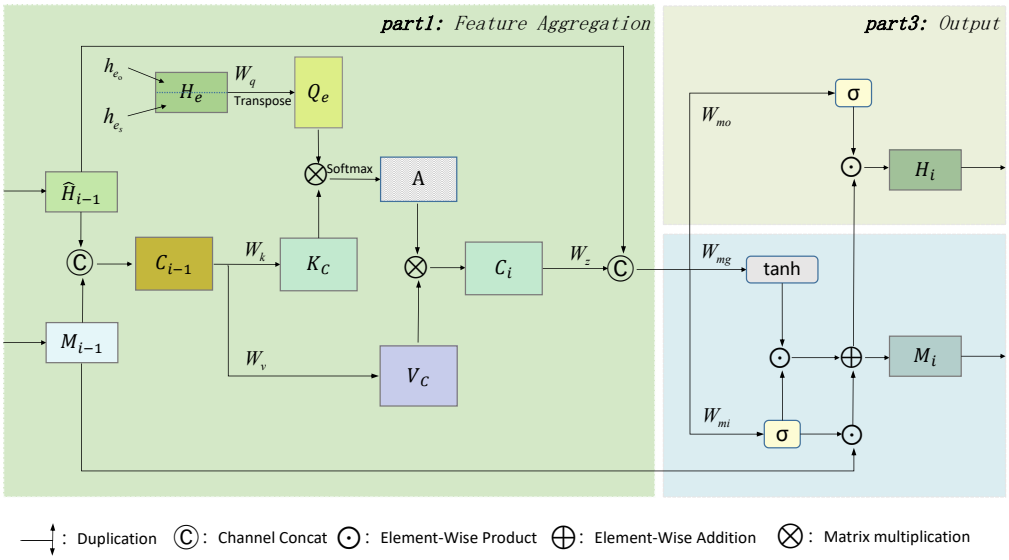

Figure 4. The proposed self-attention memory module.

3.3.1. Feature Aggregation

To match the contexts of interest for entity pairs, we first splice $\widehat{\boldsymbol{H}}_{i-1}$ and $\boldsymbol{M}_{i-1}$ into the aggregation matrix $\boldsymbol{C}_{i-1}$, and then perform a self-attentive operation on the entity pair matrix $\boldsymbol{H}_e$ with the aggregation matrix $\boldsymbol{C}_{i-1}$. First, we calculate the value matrix $\boldsymbol{C}_{i-1}$ corresponding to $\boldsymbol{V}_c = W_v\boldsymbol{C}_{i-1} \in \mathbb{R}^{C\times N}$ and the key matrix $\boldsymbol{K}_c = W_k\boldsymbol{C}_{i-1} \in \mathbb{R}^{C\times N}$. Then, we calculate the query matrix corresponding to $\boldsymbol{H}_e\boldsymbol{Q}_e = W_q\boldsymbol{H}_e \in \mathbb{R}^{C\times 2}$, where W_v, W_k, W_q are the weight parameters of the 1×1 convolution kernel. The product of the matrices is then used to compute the attention matrix $\boldsymbol{A}$ of the entity pair for the aggregated feature:

$$\boldsymbol{A} = softmax\left(\boldsymbol{Q}_e^T\boldsymbol{K}_c\right) \in \mathbb{R}^{2\times N} \tag{3}$$

The $a_{i=0,j}$ in $\boldsymbol{A}$ represents the subject's attention to the j^{th} feature vector in $\boldsymbol{C}_{i-1}$, $a_{i=1,j} \in \boldsymbol{A}$ represents the object's attention to the j^{th} feature vector in $\boldsymbol{C}_{i-1}$. To find the feature vector that both the subject/object entities are interested in, we calculate the cross-attention by multiplying the attention of both, and the normalized cross-attention weights are multiplied with the value matrix $\boldsymbol{V}_c$ to get the new aggregated feature $\boldsymbol{C}_i$. Finally, $\boldsymbol{C}_i$ is stitched together with the current moment feature as the input for the next step:

$$\begin{aligned} a^{(s,o)} &= a_{i=0,j} \cdot a_{i=1,j} \quad j \in \{1, 2, \ldots, N\} \\ a^{(s,o)} &= a^{(s,o)} / \mathbf{1}^T a^{(s,o)} \\ \boldsymbol{C}_i &= \boldsymbol{V}_c^T a^{(s,o)} \\ \boldsymbol{Z} &= W_z\left[\boldsymbol{C}_i; \widehat{\boldsymbol{H}}_{i-1}\right] \end{aligned} \tag{4}$$

3.3.2. Memory Update

In this paper, a gating mechanism similar to the GRU [26] (Gate Recurrent Unit) model is used to update the memory adaptively. The input gating i' and the fusion feature g' are generated by aggregating the features $\boldsymbol{C}_i$ and the original input $\hat{\boldsymbol{H}}_{i-1}$. To reduce the parameters, it is straightforward to use $(1 - i')$ to represent the forgetting gate, and the

specific update process is as follows. To reduce the number of parameters, we directly use $(1 - i')$ to represent the oblivion gate. The specific update process is as follows:

$$\begin{aligned} i' &= \sigma\left(W_{m;zi} * \boldsymbol{C}_i + W_{m;hi} * \widehat{\boldsymbol{H}}_{i-1} + b_{m;i}\right) \\ g' &= \tanh\left(W_{m;zg} * \boldsymbol{C}_i + W_{m;hg} * \widehat{\boldsymbol{H}}_{i-1} + b_{m;g}\right) \\ \boldsymbol{M}_t &= \left(1 - i'\right) \circ \boldsymbol{M}_{t-1} + i' \circ g' \end{aligned} \tag{5}$$

To further reduce the computational effort, this paper uses depthwise separable convolution [27] instead of standard convolution operations. Compared with the original storage unit updated by ConvLSTM through convolutional operations, the memory module proposed in this paper is updated not only by convolutional operations but also by aggregating features $\boldsymbol{C}_i$, which can capture the global dependencies of entities at all times. Therefore, we consider that the memory module can contain the international and contextual information filtered by the attention mechanism.

3.3.3. Output

The output of the module is the dot product of the output gate o' and the updated memory $\boldsymbol{M}_i$. The procedure is as follows:

$$\begin{aligned} o'_t &= \sigma(\boldsymbol{W}_{m;zo} * \mathbf{Z} + W_{m;ho} * \boldsymbol{H}_i + b_{m;o}) \\ \boldsymbol{H}_i &= o'_t \circ \boldsymbol{M}_t \end{aligned} \tag{6}$$

3.4. Classification Module

The entity pair embedding e_s, e_o and the contextual key feature matrix $\boldsymbol{M}^{(s,o)}$ can be obtained through the above steps for the word embedding sequence. These entities are mapped to hidden states by feedforward networks, and then, we use bilinear functions and sigmoid activation functions to calculate the probability of the relation r:

$$\begin{aligned} z_s &= \tanh\left(W_s e_s + W_{m1} \boldsymbol{M}^{(s,o)}\right) \\ z_o &= \tanh\left(W_o e_s + W_{m2} \boldsymbol{M}^{(s,o)}\right) \\ \mathrm{P}(r \mid e_s, e_o) &= \sigma(z_s W_r z_o + b_r) \end{aligned} \tag{7}$$

where $\{W_s, W_0, W_{m1}, W_{m2}\} \in \mathbb{R}^{d \times d}$ are the training parameters of the model, $p(r \mid e_s, e_o)$ is the probability of the existence of various relations between entity pairs predicted by the model. Previous work [19] observed that the entity relation extraction dataset generally has an unbalanced relation distribution and most of the entity pairs are unrelated to each other. So, inspired by the idea that the target class score is greater than every non-target class in Circle Loss [28], we introduce a balanced loss function training model. Initially, a loss function in the logsumexp form can be formulated as follows:

$$\log\left(1 + \sum_{i \in \Omega_{pos,j}, \Omega_{neg}} e^{s_i - s_j}\right) \tag{8}$$

We also note that the RE task is essentially a multi-label classification task. Therefore, a threshold is needed to determine which classes need to be output. Instead of introducing an additional threshold class as in the previous work [14], we directly use the score s_{NA}

of the unrelated class NA as the threshold, and thus, the loss function can be further obtained as:

$$\begin{aligned} &\log\left(1+\sum_{i\in\Omega_{pos},j\in\Omega_{neg}} e^{s_i-s_j}+\sum_{i\in\Omega_{pos},j\in\Omega_{neg}} e^{s_i-s_{NA}}+\sum_{i\in\Omega_{pos},j\in\Omega_{neg}} e^{s_{NA}-s_j}\right) \\ &=\log\left(e^{s_{NA}}+\sum_{i\in\Omega_{pos}} e^{s_i}\right)+\log\left(e^{-s_{NA}}+\sum_{j\in\Omega_{neg}} e^{-s_j}\right) \end{aligned} \tag{9}$$

This loss expects all target class scores to be greater than each non-target class, and also expects all target class scores to be greater than the threshold s_{NA} and all non-target class scores to be less than the threshold s_{NA}. In this way, the model can automatically learn and determine the labels to be output: output the relation classes greater than the threshold as having a relation, and otherwise output no relation label NA.

4. Experiment and Results

4.1. Dataset

We evaluated the proposed model on three popular document-level relationship extraction datasets (DocRED [4], CDR [29], and GDA [30]), all of which involve challenging relational inference on multiple entities across multiple sentences. DocRED and CDR provide human-validated data. They use crowdsourcing to manually annotate all possible relationships in the documents, which ensures the quality of the data and better guides the deep learning model. In order to ensure the validity and generality of our approach, we have compared performance primarily on the human-validated part of the DocRED dataset. The statistical information for each dataset is shown in Table 1.

Table 1. Summary of DocRED, CDR and GDA datasets.

Dataset	Train	Dev	Test	Entities/Doc	Mentions/Doc	Mention/Sent	Relation
DocRED	3053	1000	1000	19.51	26.19	3.58	96
CDR	500	500	500	6.78	19.21	2.48	1
GDA	29,192	-	1000	4.80	18.53	2.28	1

DocRED is a large-scale dataset built with Wikipedia. It provides comprehensive manual annotation of entity mentions, entity types, relationship facts, and corresponding inference-supporting evidence. In addition, DocRED collects remotely supervised data, which uses a fine-tuned BERT model to identify entities and link them to Wikidata. 101,873 document instances are then scaled by obtaining relationship labels through remote monitoring.

CDR (Chemical-Disease Reactions dataset) is a biomedical dataset constructed with PubMed abstracts. It contains 1500 documents with human annotations, which are equally divided into a training set, a development set, and a test set. CDR is a binary classification task designed to identify induced relations from chemical entities to disease entities, and is of great importance for biomedical research.

GDA (Gene-Disease Associations dataset) is similar to the CDR and is also a binary relationship classification task for identifying interactions between gene and disease concepts but at a much larger scale. Constructed using the remotely supervised MEDLINE summarization technique, it contains 29,192 documents as training set and 1000 documents as a test set.

4.2. Experimental Settings

The experiments in this paper are based on Pytorch, using BERT-base and RoBERTa-large as Encoder on DocRED, and SciBERT-base as Encoder on CDR and GDA datasets. We use AdamW and set a warm-up learning rate of 6% to optimize the model. The detailed

hyper-parameters settings are shown in Table 2. The model is trained on an NVIDIA P100 16GB GPU. We measured RE performance by calculating precision, recall and F-measurement scores on the test set. By evaluating all false positives, false negatives, correct positives and correct negatives, we used the standard formula to calculate the F1-score as:

$$\begin{aligned} \text{Precision} &= \frac{\text{correct positive predict}}{\text{all samples predicted as positive}} \\ \text{Recall} &= \frac{\text{correct positive predict}}{\text{all positive samples}} \\ F1 &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \end{aligned} \tag{10}$$

Table 2. Hyper-parameters Setting.

Method	DocRED		CDR	GDA
	BERT	RoBERTa	SciBERT	SciBERT
Batch size	4	4	4	16
Epoch	30	30	30	10
lr for encoder	5×10^{-5}	5×10^{-5}	2×10^{-5}	2×10^{-5}
lr for LSTM	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}

4.3. Results on the DocRED Dataset

We compared our model with graph neural models, including GEDA [16], LSR [7], GLRE [9], GAIN [8], HeterGSAN [18]; and transformer-based models, including BERT [19], BERT-TS [19], HIN [20], Coref [21], ATLOP [14]. We run five experiments with different random seeds and record the mean and standard deviation on the development set. The experimental results are shown in Table 3, and the SAM-LSTM-BERTbase in this paper achieved the best-known results on the DocRED dataset. These models' performance results are taken from their original articles.

Table 3. Results (%) on the development and test set of DocRED.

Model	Year	Dev		Test	
		Ign F1	F1	Ign F1	F1
GEDA-BERT$_{base}$ [16]	2020	54.52	56.16	53.71	55.74
LSR-BERT$_{base}$ [7]	2020	52.43	59.00	56.97	59.05
GLRE-BERT$_{base}$ [9]	2020	-	-	55.40	57.40
GAIN-BERT$_{base}$ [8]	2020	59.14	61.22	59.00	61.24
HeterGSAN-BERT$_{base}$ [18]	2021	58.13	60.18	57.12	59.45
BERT$_{base}$ [19]	2019	-	54.16	-	53.20
BERT-TS$_{base}$ [19]	2019	-	54.42	-	53.92
HIN-BERT$_{base}$ [20]	2020	54.29	56.31	53.70	55.60
CorefBERT$_{base}$ [21]	2020	55.32	57.51	54.54	56.96
ATLOP-BERT$_{base}$ [14]	2021	59.22	61.09	59.31	61.30
SAM-LSTM-BERT$_{base}$		**60.18 ± 0.14**	**61.96 ± 0.15**	**60.68**	**62.02**
BERT$_{large}$ [19]	2019	56.67	58.83	56.47	58.69
CorefBERT$_{large}$ [19]	2020	56.82	59.01	56.40	58.83
RoBERTa$_{large}$ [20]	2020	57.14	59.22	57.51	59.62
CorefRoBERTa$_{large}$ [21]	2020	57.35	59.43	57.90	60.25
ATLOP-BERT$_{large}$ [14]	2021	61.32	63.18	61.39	63.40
SAM-LSTM-RoBERTa$_{large}$		**62.13 ± 0.13**	**63.89 ± 0.16**	**62.25**	**64.03**

4.4. Results on the Biomedical Datasets

In experiments on two biomedical datasets, we compared SAM-LSTM with many models, including EoG [6], LSR [7], DHG [17], GLRE [9], and ATLOP [14]. We applied the

SciBERTbase [31] model pre-trained in the scientific publications corpus. The results are shown in Table 4, where SAM-LSTM improved the F1 scores of GDA 0.8%. We also note that [22] achieved better results than we did on the CDR dataset. We believe this is due to their use of a pre-trained biomedical language representation model and the injection of domain expertise in the RE process.

Table 4. Results (%) on the biomedical datasets CDR and GDA.

Model	Year	CDR	GDA
EoG [6]	2019	63.6	81.5
LSR [7]	2020	64.8	82.2
DHG [17]	2020	65.9	83.1
GLRE [9]	2020	68.5	-
SciBERT$_{base}$ [31]	2019	65.1	82.5
Kw-BioBERT [22]	2021	**76.4**	-
ATLOP-SciBERT$_{large}$ [14]	2021	69.4	83.9
SAM-LSTM-SciBERT$_{large}$		73.5	**84.7**

4.5. Ablation Study

A series of comparative experiments are conducted to validate the effectiveness of the components in this paper. This is shown in Table 5.

Table 5. Ablation study of SAM-LSTM on Dev.

Model	Ign F1	F1
w/o Memory	59.45	60.89
w/o Self-Attention	58.88	60.48
w/o Balanced loss	59.32	61.22
w/o SAM-LSTM	57.60	59.52

w/o Memory means that the model only calculates self-attention with the current input each time and does not update memory; **w/o Self-Attention** means that the standard ConvLstm operation is used to replace the Self-Attention part of the model; **w/o Balanced loss** means that the loss function of the baseline is used. In addition, considering that the BERT model also comes with a multi-headed attention mechanism, we directly remove the LSTM layer and test the effect of the BERT+classifier.

The result reveals that the performance of the model decreases after removing any of the components. The Memory module and Self-Attention have the greatest impact, bringing a decrease of 1.07% and 1.48% after removal. This indicates the effectiveness of the SAM-LSTM network for extracting document-level entity relations. In addition, the results of LSTM removal demonstrate that directly training the attentional relations of BERT to weight the contexts cannot achieve the desired results. We suggest that this may be due to the large parameters of the pre-trained model, and the training corpus within the dataset cannot support the model to shift its attention to keywords through learning rapidly.

4.6. Case Study

In order to further investigate the effectiveness of this model compared to the ATLOP model, we divided the dataset into different groups in terms of corpus length and tested the ATLOP model and our model on them separately.

From the results in Figure 5, it can be seen that the SAM-LSTM model in this paper consistently outperforms the ATLOP model. It can also be seen that the performance of our model decreases more slowly when the text length rises. This indicates that the method in this paper can preserve the key feature information even in longer difficult cases through a powerful self-attentive mechanism, while being less affected by irrelevant information. We also note that the time consumption of our model does not increase significantly when

faced with long difficult cases. This is mainly influenced by the length of the sequence due to the nature of the LSTM itself. However, the processing time of our method increased by a factor of 1.5 on average compared to ATLOP. We believe that this is mainly due to our strategy of successfully matching keywords for each pair of entities, which generates a relatively large amount of computation when there are more entities in the document.

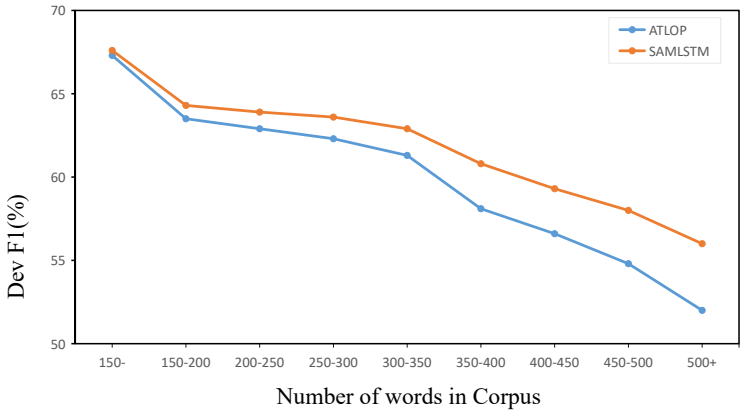

Figure 5. Results of different text lengths on dev.

In addition, to explore whether the models in this paper can actually capture key features, experiments are designed to compare the contextual feature extraction ability of the SAM-LSTM model with that of the ATLOP model. The analysis continues using the corpus mentioned in Section 1. While classifying the relation between the two entities "John Stanistreet" and "Bendigo" in the text, the contextual features used by the two models for classification are first obtained. Then, the cosine similarity is computed between the two and the word embeddings of the manually selected keywords "born" and "died".

The experimental results in Table 6 show that the similarity of our model is up to 8.13% higher than that of ATLOP. This supports why our model performs better: ATLOP uses the attention relation of the pre-trained model to weight the average word embedding of the full text, so that the key features are diluted by the large number of irrelevant vectors within the document, which deviates from the features originally needed for the downstream classification task. In contrast, the self-attentive memory unit used in this paper can capture and record these features while processing full-text information, which can achieve more favorable results for classification.

Table 6. Keyword similarity comparison.

Keyword	ATLOP	SAM-LSTM
"born"	0.6360	0.6813
"died"	0.5669	0.6482

To further validate the performance of our model in difficult cases, we chose a text containing distant relational dependencies as an example. As shown in Figure 6, when our model classifies the relationship between "Tang Dynasty" and 'Chang'an', it can be seen that the subject 'Tang Dynasty' is separated from the object 'Chang'an' by 13 sentences.

However, we can see from the similarity visualization that our model still accurately filters the keyword "capital" in 15 sentences and finally outputs features that are highly similar to it.

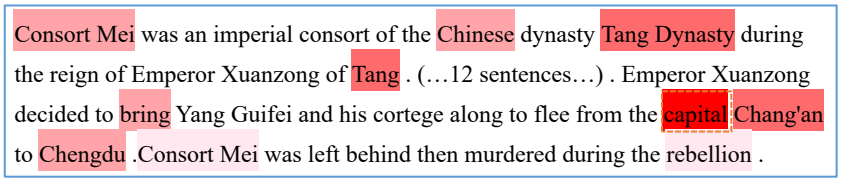

Figure 6. A visualization of the similarity of the text we selected. The darker the color of the word the more similar it is to the features we extracted.

5. Conclusions

This paper proposes the SAM-LSTM model to reconstruct global entity dependencies to find the critical semantic features. We first illustrate that it is not appropriate to use the attention layer of the pre-trained model directly. Then, we construct the new keyword attention relation and extract contextual features via recording and updating memory. Finally, we use a multi-label classification loss function that can balance the categories. With the above techniques, we further improve the performance of the document-level RE task on a public dataset. This paper also designs experiments to demonstrate that the model in this paper can better capture local keyword information compared to the current state-of-the-art models. The approach of enhancing relation classification by capturing the keyword features of entity relation dependencies is effective. Since we have achieved advanced results on both open and specialized domains/datasets, we can say that the approach of this paper can be applied to the domains not exposed to the model. Without constructing keyword features for each specific domain, our model has strong applicability to the new domains. In the course of our experiments, we also found that processing the context for each entity pair is relatively time-consuming. We will explore a common keyword extraction method for documents in the future to speed up the computation of the model and allow all entity pairs to find the content of interest in a shared keyword pool.

Author Contributions: Conceptualization, H.K. and H.C.; methodology, H.K; experiment, H.C.; validation, X.M.; data analysis, H.C. and X.M.; original draft preparation, X.M. and X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No.61772088).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Webber, B.; Cohn, T.; He, Y.; Liu, Y. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020.
2. Feng, J.; Huang, M.; Zhao, L.; Yang, Y.; Zhu, X. Reinforcement learning for relation classification from noisy data. In Proceedings of the AAAI Conference on Artificial Intelligence, Hilton New Orleans Riverside, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
3. Yao, Y.; Ye, D.; Li, P.; Han, X.; Lin, Y.; Liu, Z.; Liu, Z.; Huang, L.; Zhou, J.; Sun, M. DocRED: A large-scale document-level relation extraction dataset. *arXiv* **2019**, arXiv:1906.06127.
4. Cheng, Q.; Liu, J.; Qu, X.; Zhao, J.; Liang, J.; Wang, Z.; Huai, B.; Yuan, N.J.; Xiao, Y. HacRED: A Large-Scale Relation Extraction Dataset Toward Hard Cases in Practical Applications. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online Event, 1–6 August 2021; pp. 2819–2831.
5. Peng, N.; Poon, H.; Quirk, C.; Toutanova, K.; Yih, W.T. Cross-sentence n-ary relation extraction with graph lstms. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 101–115. [CrossRef]
6. Christopoulou, F.; Miwa, M.; Ananiadou, S. Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. *arXiv* **2019**, arXiv:1909.00228.

7. Nan, G.; Guo, Z.; Sekulić, I.; Lu, W. Reasoning with latent structure refinement for document-level relation extraction. *arXiv* **2020**, arXiv:2005.06312.
8. Zeng, S.; Xu, R.; Chang, B.; Li, L. Double graph based reasoning for document-level relation extraction. *arXiv* **2020**, arXiv:2009.13752.
9. Wang, D.; Hu, W.; Cao, E.; Sun, W. Global-to-local neural networks for document-level relation extraction. *arXiv* **2020**, arXiv:2009.10359.
10. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
11. Clark, K.; Khandelwal, U.; Levy, O.; Manning, C.D. What does bert look at? An analysis of bert's attention. *arXiv* **2019**, arXiv:1906.04341.
12. Tenney, I.; Das, D.; Pavlick, E. BERT rediscovers the classical NLP pipeline. *arXiv* **2019**, arXiv:1905.05950.
13. Su, S.Y.; Chuang, Y.S.; Chen, Y.N. Dual Inference for Improving Language Understanding and Generation. *arXiv* **2020**, arXiv:2010.04246.
14. Zhou, W.; Huang, K.; Ma, T.; Huang, J. Document-level relation extraction with adaptive thresholding and localized context pooling. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 2–9 February 2021; Volume 35, pp. 14612–14620.
15. Xingjian, S.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.C. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*; Hong Kong Observatory: Hong Kong, China, 2015; pp. 802–810.
16. Li, B.; Ye, W.; Sheng, Z.; Xie, R.; Xi, X.; Zhang, S. Graph enhanced dual attention network for document-level relation extraction. In Proceedings of the 28th International Conference on Computational Linguistics, Raffles Boulevard, Pan Pacific Singapore, Singapore, 25–27 March 2020; pp. 1551–1560.
17. Zhang, Z.; Yu, B.; Shu, X.; Liu, T.; Tang, H.; Yubin, W.; Guo, L. Document-level Relation Extraction with Dual-tier Heterogeneous Graph. In Proceedings of the 28th International Conference on Computational Linguistics, Raffles Boulevard, Pan Pacific Singapore, Singapore, 25–27 March 2020; pp. 1630–1641.
18. Xu, W.; Chen, K.; Zhao, T. Document-level relation extraction with reconstruction. In Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI-21), Vancouver, BC, Canada, 2–9 February 2021.
19. Wang, H.; Focke, C.; Sylvester, R.; Mishra, N.; Wang, W. Fine-tune bert for docred with two-step process. *arXiv* **2019**, arXiv:1909.11898.
20. Tang, H.; Cao, Y.; Zhang, Z.; Cao, J.; Fang, F.; Wang, S.; Yin, P. Hin: Hierarchical inference network for document-level relation extraction. *Adv. Knowl. Discov. Data Min.* **2020**, *12084*, 197.
21. Ye, D.; Lin, Y.; Du, J.; Liu, Z.; Li, P.; Sun, M.; Liu, Z. Coreferential reasoning learning for language representation. *arXiv* **2020**, arXiv:2004.06870.
22. Zhu, X.; Zhang, L.; Du, J.; Xiao, Z. Full-Abstract Biomedical Relation Extraction with Keyword-Attentive Domain Knowledge Infusion. *Appl. Sci.* **2021**, *11*, 7318. [CrossRef]
23. Lv, X.; Lv, X.; You, X.; Dong, Z.; Han, J. Relation extraction toward patent domain based on keyword strategy and attention+ BiLSTM model (short paper). In *Collaborative Computing: Networking, Applications and Worksharing*; Springer: Berlin, Germany, 2019; pp. 408–416.
24. Lin, Z.; Li, M.; Zheng, Z.; Cheng, Y.; Yuan, C. Self-attention convlstm for spatiotemporal prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, Hilton New York Midtown, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 11531–11538.
25. Jia, R.; Wong, C.; Poon, H. Document-Level *N*-ary Relation Extraction with Multiscale Representation Learning. *arXiv* **2019**, arXiv:1904.02347.
26. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
27. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; pp. 1251–1258.
28. Sun, Y.; Cheng, C.; Zhang, Y.; Zhang, C.; Zheng, L.; Wang, Z.; Wei, Y. Circle loss: A unified perspective of pair similarity optimization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–18 June 2020; pp. 6398–6407.
29. Li, J.; Sun, Y.; Johnson, R.J.; Sciaky, D.; Wei, C.H.; Leaman, R.; Davis, A.P.; Mattingly, C.J.; Wiegers, T.C.; Lu, Z. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database* **2016**, *2016*, 10. [CrossRef] [PubMed]
30. Wu, Y.; Luo, R.; Leung, H.C.; Ting, H.F.; Lam, T.W. Renet: A deep learning approach for extracting gene-disease associations from literature. In *International Conference on Research in Computational Molecular Biology*; Springer: Berlin, Germany, 2019; pp. 272–284.
31. Beltagy, I.; Lo, K.; Cohan, A. Scibert: A pretrained language model for scientific text. *arXiv* **2019**, arXiv:1903.10676.