electronics

MDPI

Article

Entity–Relation Extraction—A Novel and Lightweight Method Based on a Gate Linear Mechanism

Guangming Peng and Xiong Chen *

Department of Electronic Engineering, Fudan University, Shanghai 200433, China; 18210720039@fudan.edu.cn
* Correspondence: chenxiong@fudan.edu.cn

Received: 8 September 2020; Accepted: 29 September 2020; Published: 4 October 2020

Abstract: Entity–relation extraction has attracted considerable attention in recent years as a fundamental task in natural language processing. The goal of entity–relation extraction is to discover the relation structures of entities from a natural language sentence. Most existing models approach this task using recurrent neural nets (RNNs); however, given the sequential nature of RNNs, the states cannot be computed in parallel, which slows the machine comprehension. In this paper, we propose a new end-to-end model based on dilated convolutional units and the gate linear mechanism as an alternative to those recurrent models. We find that relation extraction becomes more difficult as the sentence length increases. In this paper, we introduce dynamic convolutions based on lightweight convolutions to process long sequences, which thus reduces the number of parameters to a low level. Another challenge in relation extraction is relation spans potentially overlapping in a sentence, representing a bottleneck for the detection of multiple relational triplets. To alleviate this problem, we design an entirely new prediction scheme to extract relational pairs and additionally boost performance. We conduct experiments on two widely used datasets, and the results show that our model outperforms the baselines by a large margin.

Keywords: entity–relation extraction; triplet; overlap; gate mechanism; dilated convolution

1. Introduction

Entity–relation extraction is a fundamental task in information extraction that aims to detect a list of triplets including two entities and the semantic relations between them from a portion of unstructured text. An example is shown in Figure 1. To date, conventional methods [1] mainly regard this task as one of relation classification after the entities are specified, ignoring the extraction of entities. These methods are therefore unable to fully exploit the rich information in the text. A more effective strategy [2] is to extract the entities first and then predict their relations; however, this ignores the underlying dependencies of entity identification and relation prediction [3,4]. To tackle this issue, joint learning frameworks have been proposed [3,5–7]. They have produced more accurate performance than previous models in this task; however, they require complicated feature engineering and rely heavily on other pre-existing natural language processing (NLP) tools, which might propagate errors [8]. Deep learning is being increasingly applied to the task of relation extraction.

Despite the promising results, several issues remain with traditional methods [8–11]. Firstly, these models are dominated by recurrent neural nets (RNNs), such as long short-term memory networks (LSTMs) [12] and gated recurrent units (GRUs) [13], which have been proven to be successful. However, due to the sequential nature of RNNs, parallel computation within a sequence is prevented [14]. Recently, convolutional neural networks (CNNs) have gained importance in natural language processing, and the first fully convolutional model for sequence-to-sequence learning was proposed by Gehring et al. [14]. Inspired by Wu's [15] work, we used the gated linear dilated residual network (GLDR) as an alternative to RNNs, which is also its first applied to the relational extraction task.

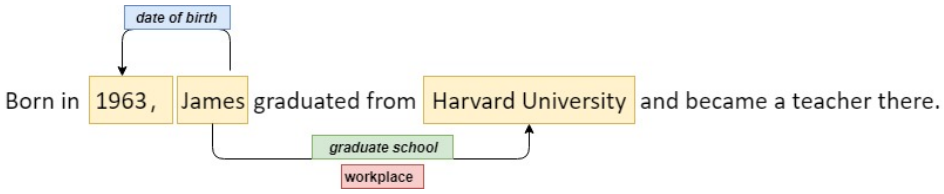


Figure 1. An example of entity–relation extraction. There are three triplets in the sentence, where <James, graduate school, Harvard University> and <James, workplace, Harvard University> have overlapping entity pairs, while both of them have a entity overlap with <James, date of birth, 1963>.

Secondly, most existing methods pay little attention to the effect of sequence length on performance. Generally, self-attention is an effective mechanism for assigning context words or characters with attention weights that define a weighted sum over context representations [16]. Attention weights are computed for all pairs of elements in the source sequence. One problem of self attention is that long sequence processing becomes very challenging because of the quadratic complexity in the sequence length. To solve this problem, researchers have proposed different kinds of attention mechanisms, including the local attention mechanism [17] and the hard attention mechanism [18]. Even though these methods improved the performance of attention mechanisms, they are either not flexible enough or require reinforcement learning for training due to their discrete nature.

Thirdly, triplet overlap is a complicated problem in entity–relation extraction [8]. Therefore, researchers proposed the copy mechanism to jointly extract entity pairs according to different decoding steps. The novel tagging scheme [9] is still unable to solve this problem completely since it only assigns a tag to each word in the sentence. In the model presented by Zheng [9], an entity belongs to at most one triple and cannot be predicted correctly when entity pairs overlap. Some recent work [19–21] also did not pay much attention to this issue as well. For example, Tran et al. [20] use unsupervised relation extraction (URE) to induce relation types, but it doesn't work when there are multiple relationships between two entities.

To effectively overcome the aforementioned challenges, in this paper, we propose an end-to-end model based on a gated linear mechanism network and dynamic convolution to tackle the task of entity–relation extraction. For the sake of clarity, we use (E1, R, E2) to represent a triplet. In general, our model consists of two parts: E1 prediction and multi-turn E2 prediction. Firstly, the encoder converts the input sentence into a fixed-length vector, where a 12-layer GLDR and dynamic convolutions are used. In this step, we need to extract all of the E1s of the sentence and place them into a "bag". Then, we sample an E1 from the bag and encode it with a bidirectional $LSTM(BiLSTM)$ layer. This side information is used to help us to predict E2s and the relations between them. Particularly, for each predefined relation, there is a corresponding prediction regarding the position of E2. In other words, we can predict both E2s and relations simultaneously and handle a situation in which relation overlap occurs.

The main contributions of our work are as follows:

- We propose a method based on a dilated convolution neural network to jointly extract entities and relations. This is the first time that dilated convolution has been used for this task. To solve the problem of the vanishing gradient, we introduce gating mechanisms. The experimental results show that this structure is more effective than RNNs and normal convolutions.
- Based on this framework, we introduce and improve dynamic convolution and overcome the problem of the worsening performance as the sentence length increases. We use a new scheme to tag the entities that is able to handle sentences with different entity overlap degrees.
- We conduct experiments on two widely used datasets—NYT and WebNLG. The experimental results show that our model outperforms the baselines with significant improvements in F1 scores.

In this paper, we measure the results with standard precision (Prec), recall (Rec), and the F1 score (F1). Experimental results show significant improvements over baseline methods, indicating that our

method is effective. The remainder of this paper is organized as follows: Section 2 briefly introduces related work and the background. We detail our model in Section 3 and provide the experimental results in Section 4. Section 5 discusses the performance of different methods. Our conclusion is discussed in Section 6.

2. Related Work

2.1. Distantly Supervised Relation Extraction

The distance supervision method [22,23] is used to automatically annotate large-scale datasets by mapping relations in a knowledge base to text; it has been successfully used in relation extraction (RE) tasks. Distance supervision assumes that sentences that contain the same entity pairs express the same relationships. For example, in the sentence "The leader of Aarhus (E1) is Jacob Bundsgaard (E2)", the entity pair "Aarhus (E1)" and Jacob "Bundsgaard (E2)" represent the relation of "LeaderName". Although distant supervision benefits from automatically generating new training data, serious mislabelling issues impact its performance. To tackle this problem, Huang et al. [24] proposed a collaborative curriculum learning (CCL) method with self-attention enhanced CNNs, and the architecture is shown in Figure 2.

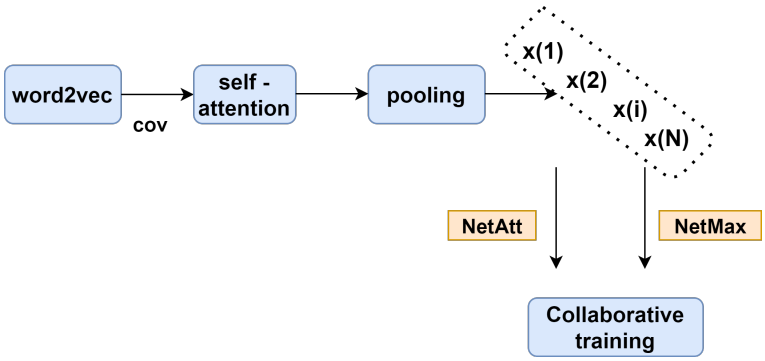


Figure 2. The overall architecture of the collaborative curriculum learning method.

Using the same sentence representation, the NetAtt and NetMax methods select sentences separately, and the conflicts between them are used to form a conflict loss so that they can regularize each other. This approach can significantly reduce noise.

2.2. Hybrid Dilated Convolution

Dilated convolution was originally developed for the computation of the wavelet decomposition algorithm "atrous" [25]. In recent years, dilated convolution networks [26] have been widely used in tasks such as semantic image segmentation [27], object detection [28], and audio generation [29]. Considering 1D signals, an r dilated convolution of x can be described as:

$$y(i) = \sum_{l=1}^{L} x(i + rl)h(l), \tag{1}$$

where $x(i)$ shows the input signals, $y(i)$ is the output with respect to x, and $h(l)$ denotes the filter of the length of L. In normal convolution, $r = 1$.

The compelling advantage of dilated convolution is that the receptive field of ConvNet grows exponentially with the network depth and soon encompasses a long sequence, considerably shortening computation paths. However, one inherent issue in dilated convolution is gridding, which worsens as the rate of dilation increases. To address this problem, Wang et al. [30] proposed a hybrid dilated convolution (HDC) framework that aids in the proper choice of the rate of dilation. See Figure 3 for an illustration of $r = [1, 2, 5]$.

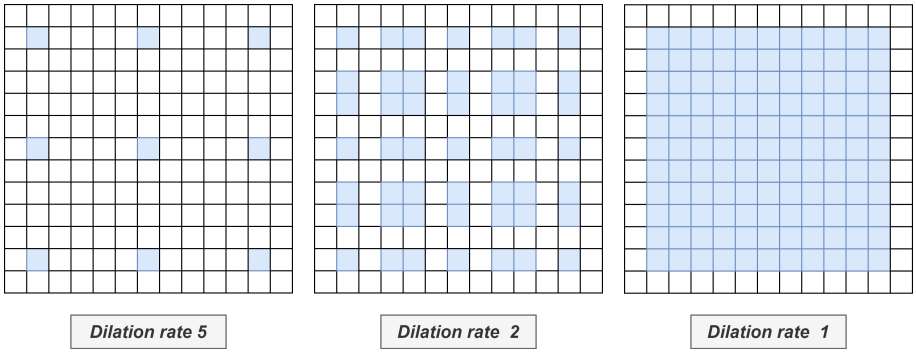


Figure 3. Illustration of the hybrid dilated convolution (HDC) framework. Left to right, convolutional layers have dilation rates of r = 5, 2, 1, respectively.

In the convolutional Bi-Directional Attention Flow (BIDAF) [15], a 17-layer GLDR with dilations of 1, 2, 4, 8, and 16 in the first five residual blocks was used. However, according to HDC, the dilation rates within a group should not have a common factor relationship. Thus, in our model, three layers with dilations of 1, 2, and 5 are grouped together, and we repeat this process three times. Finally, we set the last layers as standard convolutions for further refinement because the receptive field is sufficiently wide.

2.3. Gated Linear Unit

The gating mechanism plays an important role in recurrent neural networks by controlling the path through which information flows in the network [12]. In contrast to RNNs, CNNs do not need forget gates; therefore, Dauphin et al. [31] proposed a novel gating mechanism, called gated linear units (GLUs), that reduces the vanishing gradient problem by providing a linear path while retaining non-linear capabilities.

Given a sequence of N words $X = (x_1, \cdots, x_i, \cdots, x_N)$ and the embedding $E = \left(w_{x_1}, \cdots, w_{x_i}, \cdots, w_{x_N}\right)$, the hidden layers $h_1, \cdots, h_l, \cdots, h_L$ are defined as:

$$h(X) = (XW + b)\sigma(XV + c), \tag{2}$$

where X is either word embedding or the output from previous layers and W, V, b, c are learned parameters. The architecture is shown in Figure 4.

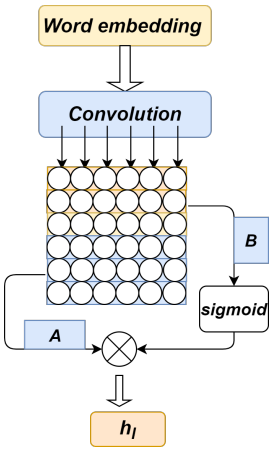


Figure 4. The structure of the gated linear units (GLUs).

In our model, we chose dilated convolution as an alternative to normal convolution, and experiments showed that this method produces more accurate performance.

3. Our Approach

In this section, we introduce a novel end-to-end model based on GLDR to jointly extract entities and their relations (E1, R, E2). This algorithm contains two stages: E1 prediction and multi-turn E2 prediction (the relations will be predicted together with E2 in the second stage). The overall structure of our model is shown in Figure 5.

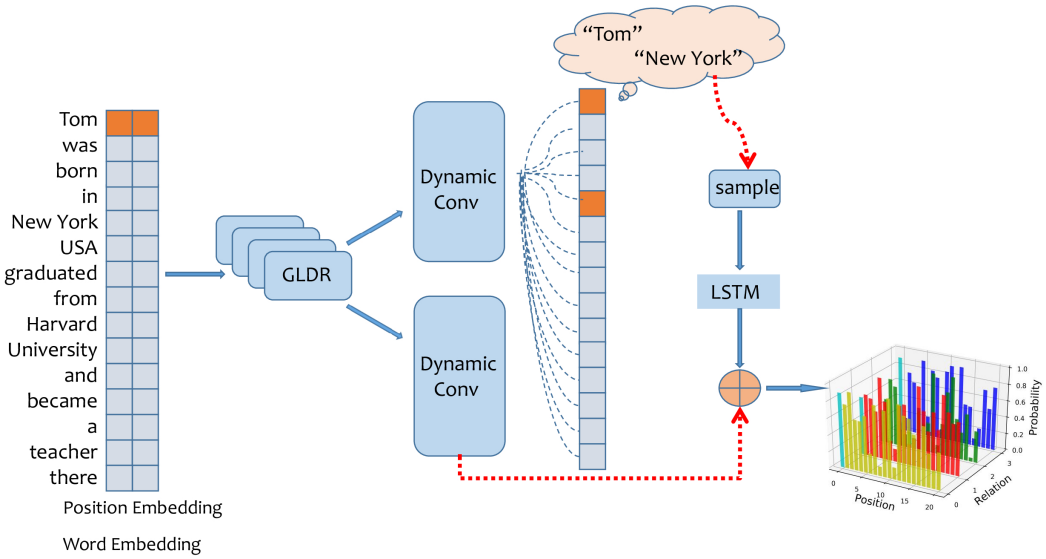


Figure 5. The overall architecture of our model. Our algorithm contains two stages, E1 prediction and multi-turn E2 prediction, and the relations are predicted together with E2 in the second stage.

3.1. E1 Prediction

3.1.1. Gated Linear Dilated Residual

To begin, we need to transform a variable-length sentence into a fixed-length vector. Given a sentence $S = (w_1, \cdots, w_i, \cdots, w_n)$, where w_i denotes the ith word in S, we embed it as a matrix X. We combine position information with word embedding, as it is useful in our architecture. The gated linear dilated residual (GLDR) used in our model is not the same as that used in Wu et al. [15]. Instead of increasing the dilation rate exponentially, we set it as $(1, 2, 5, 1, 2, 5, 1, 2, 5, 1, 1, 1)$; the reason for this is detailed in Section 2. Similar to ResNet [32], we sum the output of GLU and the input X:

$$M = X + (XW + b)\sigma(XV + c), \tag{3}$$

where X is the input from the previous layers and W, V, b, c are learned parameters. Note that after learning in several layers, the receptive field (RF) is sufficiently wide for the task, so we set the dilation rate to 1 for the last three layers for further refinement.

3.1.2. Dynamic Convolution

In this paper, we abandon the idea of using a self-attention mechanism and introduce multi-channel integrated dynamic convolutions based on lightweight convolutions. Unlike the dynamic convolution (DC) method previously proposed [33], we reduce the number of parameters by sharing weights in several forms and integrating information from different channels. Our approach also uses a function to predict convolution kernels at every time step to improve performance.

In this subsection, we introduce the details of multi-channel integrated dynamic convolution (MCIDConv). However, we first need to briefly describe depthwise and lightweight convolution. Depthwise convolution can reduce the number of parameters from d_2k to dk by performing a convolution

independently over every channel, where d is the input and output dimension and k is the kernel width. We describe the output of a depthwise convolution $O \in R^{n \times d}$ as:

$$DepthwiseConv(X, W_{c,:}, i, c) = \sum_{j=1}^{k} W_{c,j} X_{(i+j-\lceil \frac{k+1}{2} \rceil)}, \tag{4}$$

where i is the index of elements and c is the output dimension. To further reduce the number of parameters, Wu et al. [33] proposed lightweight convolutions that share certain output channels. The output is calculated as follows:

$$O_{i,c} = DepthwiseConv\Big(X, softmax\Big(W_{\lceil \frac{cH}{d} \rceil,:}\Big), i, c\Big). \tag{5}$$

This approach ties the parameters of every subsequent number of $\frac{d}{H}$ channels so that the number of parameters can be reduced to Hk. Dynamic convolutions build on this process using a function to predict the convolution kernel at every time-step. It is computed as:

$$DC(X, i, c) = LightConv\Big(X, f(X_i)_{h,:}, i, c\Big). \tag{6}$$

Although this method reduces the number of parameters to a very low level and models equipped with dynamic convolutions can be competitive with state-of-the-art self-attention models, some problems remain. One issue is that parameter H is determined empirically, and if k is too large, sharing weights can lead to a loss of information. Depthwise convolution-based computation does not effectively use the information of different maps in the same spatial position. Therefore, we propose a multi-channel integrated convolution based on DynamicConv with the aim of maintaining a balance between parameter reduction and information retention. Figure 6 illustrates multi-channel integrated convolution, whereas pointwise convolution is used in our method, which can be defined as follows:

$$MCDynamicConv(X, i, c) = pointwiseConv\Big(LightConv\Big(X, f(X_i)_{h,:}, i, c, H_i\Big)\Big). \tag{7}$$

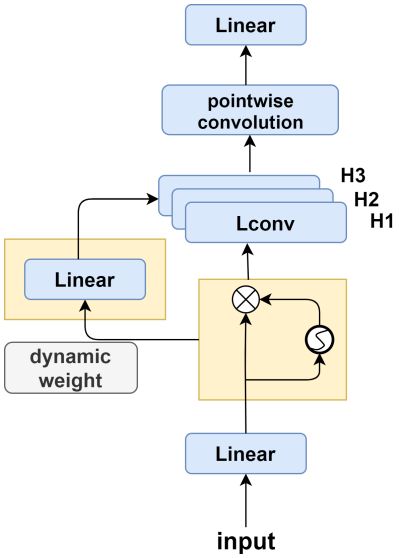


Figure 6. Illustration of multi-channel integrated dynamic convolutions.

We set three different values of H to share parameters between the channels so that each channel has the opportunity to be trained with different parameters; a pointwise convolution is used to transform the dimensions. We also apply a pointwise convolution to the output of dynamic convolution to integrate information from other identical spatial locations.

3.1.3. Prediction

We use a fully connected layer to detect the position of E1. The position vector $p = (p_1, \cdots, p_i, \cdots, p_n)$ is calculated as follows:

$$p_i = sigmoid(selu(wz_i + b)), \tag{8}$$

where w is the weight matrix, n is the length of the sentence, b is the bias, $selu(\cdot)$ is the activation function [34], and z_i is the output of dynamic convolution. There may be more than one E1 in a sentence; for example, if $p = (1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$, the sentence contains two E1s and their position. Then, we can extract all of them to form a bag. Each E1 of the bag' will be encoded as side information to predict the E2s and the relations between them.

3.2. Multi-Turn E2 Prediction

To predict E2, we first need to encode E1 using bi-directional RNNs. We denote $E = (e_1, \cdots, e_L)$ as the embedding of E1, which is sampled from the results of the first step. For the forward RNNs, output o_l^{E1} and the hidden state $h_l^{E1} (l < L)$ are defined as follows:

$$\overrightarrow{o}_1^{E1}, \overrightarrow{h}_l^{E1} = f\left(e_l, \overrightarrow{h}_{l-1}^{E1}\right), \tag{9}$$

where $f(\cdot)$ is the coder function. Similarly, we can obtain the backward RNN output $\left(\overleftarrow{o}_L^{E1}, \cdots, \overleftarrow{o}_1\right)$ and $\left(\overleftarrow{h}_L^{E1}, \cdots, \overleftarrow{h}_1\right)$. Then, a concatenation of forward and backward RNN hidden states $E_o = \left[\overrightarrow{h}_l^{E1}, \overleftarrow{h}_l^{E1}\right]$ is used to represent E1 and important side information to help the model predict E2. We obtain M from the previous layer and feed it into another convolution layer to obtain the output $Z^{'}$. Similar to E1 prediction, we train our model to calculate the position of E2 in the sentence. Suppose there are T valid relations in total; for each relation, we calculate the position of E2. Practically, several kinds of relations may exist between a pair of entities. The output can be described as:

$$relation \begin{cases} p_1 = sigmoid(selu(w_1 u + b_1)) \\ \cdots\cdots \\ p_t = sigmoid(selu(w_t u + b_t)) \\ \cdots\cdots \\ p_T = sigmoid(selu(w_T u + b_T)) \\ , \end{cases} \tag{10}$$

where u is the concatenation of $Z^{'}$ and E_o and t denotes different kinds of relations. For each E1, we detect whether there is a corresponding E2 in the sentence and predict their relations. As such, all the triplets can be extracted, including overlapping triplets.

3.3. Loss Function

Focal loss [35] is a loss function applied to address the issue of the class imbalance problem. It was originally used to learn the hard examples that prevailed in one-shot object detectors. The cross-entropy (CE) function and focal loss function are defined as follows:

$$L = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}), \tag{11}$$

$$L = -y\alpha(1 - \hat{y})^{\gamma} \log(\hat{y}) - (1 - y)(1 - \alpha])\hat{y}^{\gamma} \log(1 - \hat{y}), \tag{12}$$

where y is the ground-truth class, $\hat{y}$ specifies the model's estimated probability, and α and γ are hyperparameters that can be used with the CE loss function for cross-validation. Mathematically, a modulation term is applied to the cross-entropy loss function. When classical cross-entropy is used as the loss function, a large amount of easily distinguishable background data occupies most of the weight of the loss function, which, to some extent, prevents the gradient from moving in a direction that is beneficial to the mining of more hard samples. Focal loss puts more weight on the hard examples and decreases the impact of easy correct predictions, making it efficient and easy for the model to learn hard examples.

4. Experiments

4.1. Datasets

We conducted experiments on two widely used datasets—New York Time (NYT) and WebNLG [36].

NYT consists of 1.18 million sentences extracted from news articles and contains 24 relations. It was developed by the distant supervision method [37], which can obtain large-scale data without performing manual labeling. In this study, we filtered the sentences that contained no positive triplets, and 64,216 sentences remained. Through random selection, the dataset was split into a training set, a test set, and a validation set.

The second dataset was WebNLG, which was originally created to promote the development of natural language generation (NLG). It contains 25,298 data, text pairs; texts including a group of triplets are sequences of one or more standard sentences. Due to all triplets being found in the standard sentence, we only needed to select one of them and ignored incomplete sentences. In our experiments, we created a valid set by randomly sampling 10% of the data from the original training set. In total, the training set contained 4500 examples and the test set contained 700.

The average numbers of triplets in each sentence are 2.98 and 1.68, and the maximum numbers of triplets are 7 and 26, respectively. The number of triplets in WebNLG dataset sentences is more evenly distributed, as shown in Figure 7.

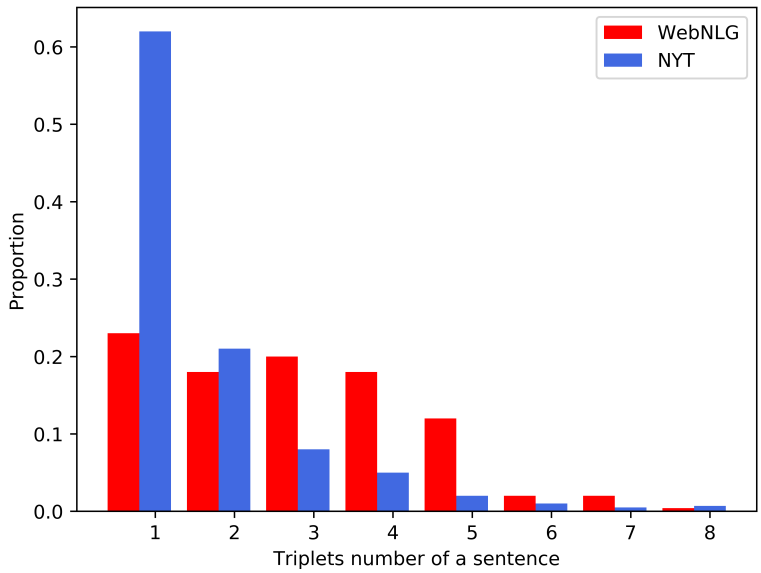


Figure 7. Distribution of the number of triplets in two datasets.

4.2. Settings

We used dilated convolutional units and and set (1, 2, 5, 1, 2, 5, 1, 2, 5, 1, 1, 1) as the dilation rate based on Wang's [30] suggestion. The word embeddings were initialed by running Word2vec, and the dimension was set to 128, the learning rate was 0.001, and the batch size was 32. The Adam

method [38] was used to optimize parameters. We used dropout on embedding layers to regularize our network, and the dropout ratio was set to 0.25.

4.3. Evaluation and Baselines

Following previous work [8], we adopted the standard precision (Prec), recall (Rec), and F1 score to evaluate the performance of each method. Precision is the ratio of correctly predicted positive samples to the total predicted positive samples. Recall is the ratio of correctly predicted positive samples to the all samples in actual class. The F1-score is the harmonic average of Precision and Recall. Their calculation formulas are as follows:

$$Precision = \frac{TP}{TP + FP}, \tag{13}$$

$$Recall = \frac{TP}{TP + FN}, \tag{14}$$

$$F1 = \frac{2PrecisionRecall}{Precision + Recall}. \tag{15}$$

where TP refers to the correctly predicted positive samples, TN refers to the correctly predicted negative samples, FP refers to when actual sample is negative, the predicted class is positive sample, FN refers to when actual class is positive, the predicted class is negative sample. A triplet was regarded as correct only when its relations and entities were both correctly predicted. We ran each experiment five times and took the average as the final result.

We compared our method with the copy mechanism and novel tagging methods [8,9] that previously exhibited the best performance. In addition, we compared the model with the recently popular distance supervision method collaborative curriculum learning (CCL) [24]. The novel tagging method uses a tagging scheme to convert the extraction to a tagging problem. The MultiDecoder is a framework based on Seq2Seq learning with a copy mechanism for multiple relational fact extraction. This model represents a sentence as a fixed-length vector first and then uses multiple decoders to decode all triplets separately, which is effective especially when triplets overlap in multi-relational extraction. We directly used the source code of the above baselines to acquire results for the same dataset.

4.4. Experimental Results

In this section, we report the experimental results of different methods on the NYT and WebNLG datasets. Table 1 compares the Prec, Rec, and F1 scores of the copy mechanism model, novel tagging model, and our model.

Table 1. Comparison of results of our model and baselines in New York Time (NYT) and WebNLG datasets.

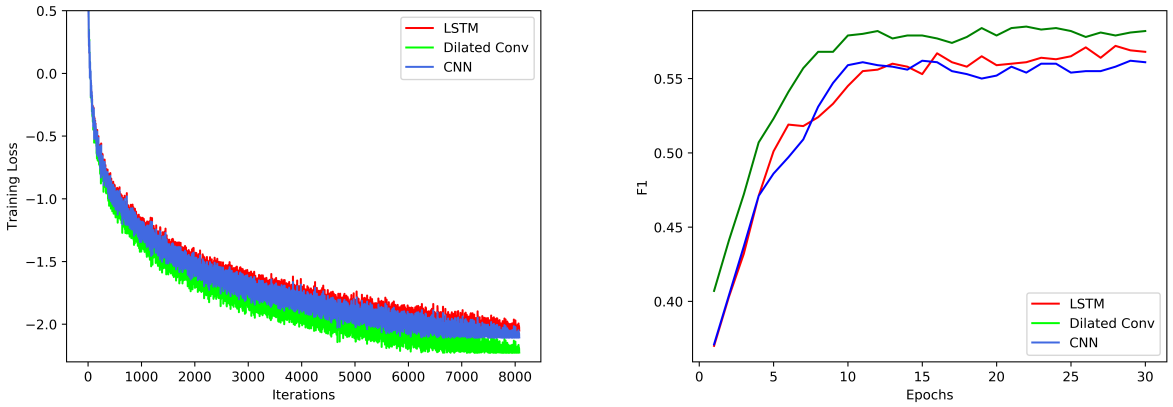
Model	NYT			WebNLG		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Copy Mechanism	0.581	0.569	0.575	0.379	0.362	0.370
Novel Tagging	0.641	0.352	0.454	0.531	0.204	0.295
CCL	0.632	0.241	0.349	0.352	0.196	0.252
Our Model	0.674	0.607	0.639	0.688	0.504	0.582

As shown above, our proposed model outperforms the baseline methods on both the NYT and WebNLG datasets and produced improvements of 0.064 and 0.212, respectively, in terms of the F1 score over the copy mechanism method. These observations verify the effectiveness of our proposed model. The performance of the CCL method is not very good. The reason may be that it is not specifically designed to solve the problem of triplet overlap. As the number of triplets in a sentence increases,

it is difficult for the CCL method to extract them all. We also observed that for the WebNLG dataset, the novel tagging method and copy mechanism do not perform well. We think that the main reason for the relatively poor performance of the baseline methods lies in the structures of the models and properties of the dataset. In the WebNLG dataset, the number of sentences of EntityOverlap accounts for a large proportion of the total, and the novel tagging model thus experiences difficulty with the dataset as it assumes that an entity only belongs to a triplet. In contrast, our model considers every relation, meaning that an entity can belong to several triplets. Further experiments proved the accuracy of this hypothesis.

4.5. Effect of Neural Network Unit

In this subsection, we compare the effect of different neural network units. For fair comparison, we replaced the dilated convolution with normal CNNs and LSTM, separately, and did not remove the other improvements. The results are shown in Figure 8.



(**a**) (**b**)

Figure 8. Comparison of results of our model based on different neural units on the WebNLG dataset. (**a**) Loss curves of our model based on different neurals units; (**b**) The value of F1 in each epoch of our model based on different neurals units.

We observed that Dilated Conv outperforms LSTM and CNN and that these two models have comparable performance. In contrast, the receptive field of the dilated ConvNet can grow exponentially with the network depth and soon encompasses a long sequence, which can help capture long dependencies. Intuitively, LSTM-based encoder–decoder architectures are more suitable for modeling long-term dependencies than CNN; however, we embeded position information in our model, which can provide it with a sense of the order of the sentence. Thus, the model based on a CNN is not considerably inferior to the model based on LSTM.

4.6. Effect of Multiple Relational Triples

To analyze our model's extraction capability from sentences that contain multiple relational triplets, we divided the dataset into seven subclasses, and each subclass contained a corresponding number of triplets. The results are shown in Figure 9.

We observed that the performance of the baseline models decreased as the number of triplets increased, and that of the novel tagging model decreased more significantly. This is reasonable because extraction becomes more difficult when there are multiple relations, and the novel tagging method is more suitable for sentences with one triplet. Our model achieved similar scores when the number of relations was less than three and then decreased gradually. This demonstrated the suitability of our method for the task. The F1 scores of the above models were close to the low level when the number of relations was seven; one reason for this, as mentioned earlier, is the increasing difficulty, and another

is that the number of sentences with seven relations is small in the dataset, which may have been insufficient to train our model.

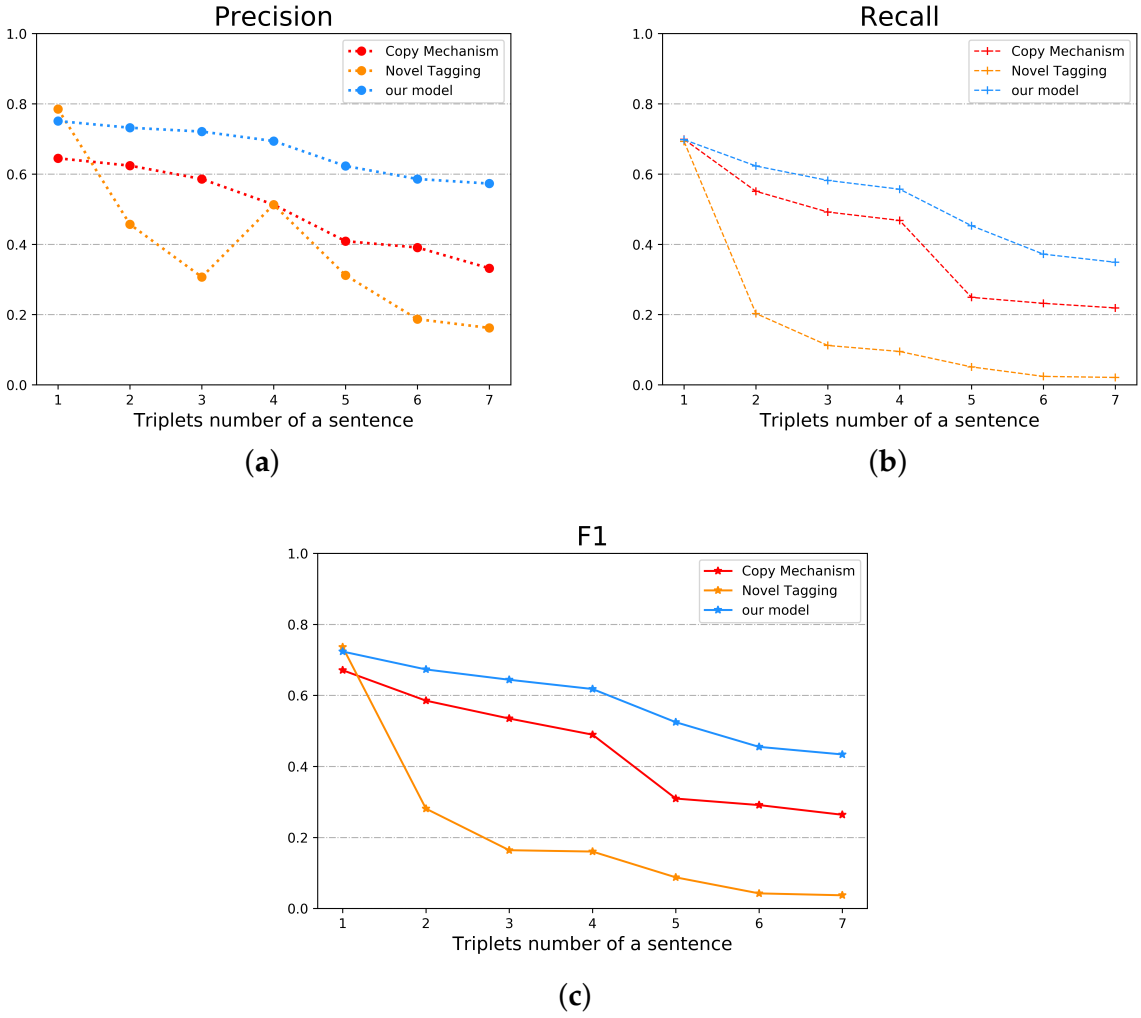


Figure 9. Comparison of results of our model and baselines on sentences that contains different number of triplets (≤ 7). (**a**) The value of precision of different methods; (**b**) The value of recall of different methods; (**c**) The value of F1 of different methods.

5. Analysis and Discussion

In this section, we focus on the performance of different methods and present a detailed analysis and explanation. We compare the prediction results of our proposed model with the copy mechanism method. Table 2 provides three representative examples extracted from the WebNLG and NYT datasets, illustrating the performance of our method and those of the baseline methods. The first column of the table is the original sentence; the second, third, and fourth columns are the extracted results of baselines and our method, respectively. In the first example, there is a normal class triplet, where the entity "Paris" is close to "France". In this case, all methods are able to predict this correctly. The second sentence is a negative example in which models may not extract entities properly; for the entity "Musee National d'Art Moderne", the result extracted by our method is "Musee" because our model can tag the entity position but fails to detect the end of the word. In this case, we think the prediction result of our model is incorrect. The result of the novel tagging model is that "Moderne" is taken as the head of the entity pair, which may be due to the influence of the principle of proximity in the method. The third example contained several relations in which entities overlapped, which increased the difficulty of detecting the entities. The novel tagging method could only extract one entity pair because it can

only divide each word into at most one triplet. Compared with the baseline methods, our model can identify more triplets when sentences have multiple relations.

Table 2. Representative results from different models. S1 is a normal class; both models extracted it correctly. S2 is a negative example. S3 represents a case in which entity pairs are overlapped.

Sentence	Copy Mechanism	Novel Tagging	Our Method
S1: Henry Louis Gates Jr., she said, "turned me on to Josephine Baker, so I headed off to France with the intention of reading her reception in Paris as a cultural text."	<Paris, location, France>	<Paris, location, France>	<Paris, location, France>
S2: Traveling from the Centre Pompidou, Musée National d'Art Moderne in Paris, this comprehensive review of the artist's drawings from the mid-1960s till now includes more than 70 works on paper.	<Paris, contains, Musee>	<Paris, contains, Moderne>	<Paris, contains, Musee>
S3: Before leaving Cairo on Wednesday, Ms. Rice met for nearly two hours with Egypt's intelligence chief, Omar Suleiman, who had traveled to Damascus, Syria, earlier this month to meet with the leaders of Hamas.	<Syria, contains, Damascus> <Egypt, contains, Cairo>	<Syria, contains, Damascus>	<Damascus, country, Syria> <Syria, contains, Damascus> <Egypt, contains, Cairo>

Although our model significantly outperformed both above-baseline approaches for both datasets, it still has some limitations. The extraction strategy of our method involves predicting E1 first and predicting "E2 + relation" jointly. For each kind of relation, we detect whether an E2 corresponds to E1 in the sentence, where E1 is sampled from the result of the first step. The advantage of this is that any entity can participate in multiple different triplets; thus, our model can handle sentences of different entity overlap degrees. But it also means that for each relationship, we need to consider whether it is appropriate. When the number of relationships is large, this will cost a lot of calculations.

6. Conclusions

In this paper, we proposed an entirely new method based on gate linear dilated convolution and investigated the end-to-end models for relational fact extraction. To solve the problem of worsening performance as the sentence length increases, we introduced dynamic convolution and thus improved the method. We also proposed a multi-turn prediction method to jointly extract relation and entity, which is effective for overlapped triplets. We conduct experiments on two widely used datasets and use Precision, Recall and F1-score to compare with some of the most advanced baselines. The results prove that our method outperforms the baselines with significant improvements.

In future work, we plan to pursue several research directions. First, we will work to increase the efficiency of our method, since some relations rarely occur and we only considered several relations with higher probability. We will adopt the approach of jointly embedding words as well as their characters, which is significant for languages that take a character as a basic unit, such as Chinese.

Author Contributions: G.P. proposed the idea, conducted the experiments, and wrote the manuscript. X.C. supervised the entire research and revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Shanghai Science and Technology Committee (STCSM) under grant number 17DZ1201605.

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

References

1. Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; Zhao, J. Relation classification via convolutional deep neural network. In Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, 23–29 August 2014; pp. 2335–2344.
2. Chan, Y.S.; Roth, D. Exploiting syntactico-semantic structures for relation extraction. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; Volume 1, pp. 551–560.
3. Li, Q.; Ji, H. Incremental joint extraction of entity mentions and relations. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, MD, USA, 22–27 June 2014; Volume 1, pp. 402–412.
4. Miwa, M.; Bansal, M. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; Volume 1, pp. 1105–1116.
5. Yu, X.; Lam, W. Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters, Beijing, China, 23–27 August 2010; pp. 1399–1407.
6. Miwa, M.; Sasaki, Y. Modeling joint entity and relation extraction with table representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1858–1869.
7. Gupta, P.; Schütze, H.; Andrassy, B. Table filling multi-task recurrent neural network for joint entity and relation extraction. In Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, 11–16 December 2016; pp. 2537–2547.
8. Zeng, X.; Zeng, D.; He, S.; Liu, K.; Zhao, J. Extracting relational facts by an end-to-end neural model with copy mechanism. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Volume 1, pp. 506–514.
9. Zheng, S.; Wang, F.; Bao, H.; Hao, Y.; Zhou, P.; Xu, B. Revisiting Unsupervised Relation Extraction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1, pp. 1227–1236.
10. Li, X.; Yin, F.; Sun, Z.; Li, X.; Yuan, A.; Chai, D.; Zhou, M.; Li, J. Entity-Relation Extraction as Multi-Turn Question Answering. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 1340–1350.
11. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
12. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
13. Cho, K.; van Merriënboer, B. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1724–1734.
14. Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; Dauphin, Y.N. Convolutional sequence to sequence learning. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Volume 70, pp. 1243–1252.
15. Wu, F.; Lao, N.; Blitzer, J.; Yang, G.; Weinberger, K. Fast Reading Comprehension with ConvNets. *arXiv* **2017**, arXiv:1711.04352.
16. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2015**, arXiv:1409.0473.
17. Yang, B.; Tu, Z.; Wong, D.F.; Meng, F.; Chao, L.S.; Zhang, T. Modeling Localness for Self-Attention Networks. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 4449–4458.

18. Indurthi, S.R.; Chung, I.; Kim, S. Look Harder: A Neural Machine Translation Model with Hard Attention. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 3037–3043.
19. Shahbazi, H.; Fern, X.; Ghaeini, R.; Tadepalli, P. Relation Extraction with Explanation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Seattle, WA, USA, 5–10 July 2020; pp. 6488–6494.
20. Tran, T.T.; Le, P.; Ananiadou, S. Revisiting Unsupervised Relation Extraction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Seattle, WA, USA, 5–10 July 2020; pp. 7498–7505.
21. Sun, T.; Tang, S.; Huang, Y.; Qian, J.; ElSherief, M.; Zhao, J.; Mirza, D.; Belding, E.; Chang, K.; Wang, W. Towards Understanding Gender Bias in Relation Extraction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Seattle, WA, USA, 5–10 July 2020; pp. 2943–2953.
22. Mintz, M.; Bills, S.; Snow, R.; Jurafsky, D. Distant supervision for relation extraction without labeled data. In Proceedings of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language, Singapore, 2–7 August 2009; pp. 1003–1011.
23. Wu, S.; Fan, K.; Zhang, Q. Improving distantly supervised relation extraction with neural noise converter and conditional optimal selector. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 7273–7280.
24. Huang, Y.; Du, J. Self-Attention Enhanced CNNs and Collaborative Curriculum Learning for Distantly Supervised Relation Extraction. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 389–398.
25. Holschneider, M.; Kronland-Martinet, R.; Morlet, J.; Tchamitchian, P. A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets*; Springer: Berlin, Germany, 1990; pp. 286–297.
26. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. *arXiv* **2016**, arXiv:1511.07122.
27. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv* **2017**, arXiv:1706.05587.
28. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. *arXiv* **2016**, arXiv:1605.06409.
29. Van den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; Kavukcuoglu, K. WaveNet: A Generative Model for Raw Audio. *arXiv* **2016**, arXiv:1609.03499.
30. Wang, P.; Chen, P.; Yuan, Y.; Liu, D.; Huang, Z.; Hou, X.; Cottrell, G. Understanding convolution for semantic segmentation. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 1451–1460.
31. Dauphin, Y.N.; Fan, A.; Auli, M.; Grangier, D. Language modeling with gated convolutional networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Volume 70, pp. 933–941.
32. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
33. Wu, F.; Fan, A.; Baevski, A.; Dauphin, Y.; Auli, M. Pay Less Attention with Lightweight and Dynamic Convolutions. *arXiv* **2019**, arXiv:1901.10430.
34. Klambauer, G.; Unterthiner, T.; Mayr, A.; Hochreiter, S. Self-normalizing neural networks. *arXiv* **2017**, arXiv:1706.02515.
35. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2999–3007.
36. Gardent, C.; Shimorina, A.; Narayan, S.; Perez-Beltrachini, L. Creating Training Corpora for NLG Micro-Planning. In Proceedings of the 55th annual meeting of the Association for Computational Linguistics (ACL), Vancouver, BC, Canada, 30 July–4 August 2017.

37. Riedel, S.; Yao, L.; McCallum, A. Modeling relations and their mentions without labeled text. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Barcelona, Spain, 20–24 September 2010; pp. 148–163.
38. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.