applied sciences

MDPI

Article

Selectively Connected Self-Attentions for Semantic Role Labeling

Jaehui Park

Department of Computer Science and Engineering, Incheon National University; South Korea; jaehui@inu.ac.kr

Received: 8 March 2019; Accepted: 19 April 2019; Published: 25 April 2019

Abstract: Semantic role labeling is an effective approach to understand underlying meanings associated with word relationships in natural language sentences. Recent studies using deep neural networks, specifically, recurrent neural networks, have significantly improved traditional shallow models. However, due to the limitation of recurrent updates, they require long training time over a large data set. Moreover, they could not capture the hierarchical structures of languages. We propose a novel deep neural model, providing selective connections among attentive representations, which remove the recurrent updates, for semantic role labeling. Experimental results show that our model performs better in accuracy compared to the state-of-the-art studies. Our model achieves 86.6 F1 scores and 83.6 F1 scores on the CoNLL 2005 and CoNLL 2012 shared tasks, respectively. The accuracy gains are improved by capturing the hierarchical information using the connection module. Moreover, we show that our model can be parallelized to avoid the repetitive updates of the model. As a result, our model reduces the training time by 62 percentages from the baseline.

Keywords: semantic role labeling; attention mechanism; selective connection

1. Introduction

Semantic representation is important for the machine to understand the meaning of data. Semantic role labeling (SRL) is a task of constructing a dependency structure to represent the semantics in natural language. In specific, it assigns pre-defined labels, called *semantic roles*, about 'when', 'who', 'what to whom' or 'where', to the non-predicate words dependent to predicates. The definition of semantic roles have been well posed in the literature [1,2]. The aim of SRL is accurately predicting the labels among verbs and other words given a sentence. SRL studies have been applied to benefit many natural language processing (NLP) applications, such as question answering [3] machine translation [4], and many others [5].

Many traditional SRL approaches have used the shallow models that rely on the syntactic features. With the recent development of deep learning [6], more and more SRL studies using deep neural networks improve state-of-the-art approaches only with large training data sets. For example, recurrent neural networks (RNN) have shown satisfactory results [7] in SRL owing to the *sequential nature* of natural language data. The variants of RNN using long short-term memory (LSTM) are popularly used in many NLP applications [8] due to their great performance in training sequential data. The LSTMs, which enable the RNNs to memorize selectively, have been regarded as viable modules that can learn the latent features over an alignment between the pair of long sentences. In SRL, several recent studies [9–11] have been adopted the LSTM-based networks to generate a sequence of labels, that is, semantic roles, corresponding to the input words. A seminal work [9] has introduced a stacked, bidirectional LSTM-based model for SRL, and has outperformed the existing studies.

However, the existing approaches [7,9–11] have several limitations. First, LSTM-based approaches overwrite a large number of parameters of the model during training. The updates of the hidden states may perform poorly when the training sequences are long. For example, the overwritten memory of an

LSTM module may not maintain the features learned at the early stage until it processes the last word. Second, due to the inherent recurrence, the current hidden state is repetitively updated by the previous hidden state. Although this process makes the models learn the dependency through time, it prevents parallel computation. Third, training with temporal dependency over the sequence may lose some spatial constraints, for example, hierarchical structure over the words in a sentence. For example, a sequence of words, construct a phrase, and a sentence is made of phrases. It is worth noting that the hierarchical representation is an effective form of natural language. Although the existing SRL studies perform greatly, there still remain challenges in training long sequences and spatial constrains, such as hierarchical information.

To address the problems, we present a novel deep architecture for SRL based on stacked attentive representations. The *attention mechanism* [12] has become a prevalent method in neural models for NLP tasks because of its impressive performance [6]. The attention mechanism works to focus the salient information over an input while generating an output. Our model is based on the self-attention mechanism [13] for sentence representation because we note that the self-attention is a proper method to capture the associations for SRL. Since the self-attention has been considered to derive the intra-connections among every word within a sentence, it has been popularly adopted in recent NLP studies [6]. The basis of our architecture is the encoder of the *transformer* [14] as in the prior work [15], which is similar to ours. However, we differ from the work [15] by stacking the self-attentions with the original module called, the *selective connection*. The module makes our model can capture the hierarchical information over the attentive representations, which are built on the self-attention modules, while training. Our intuition is that the combination of the self-attention module and the selective memory units would perform better because the former represents the correlation between words in a sentence, and the latter represents the hierarchical structure in the learned representation of the sentence. By adapting gating units in the selective connection module, our model can manage where to focus over the sentence while encoding its representations. As a result, the selective connection module can maintain latent features through the entire stacked network efficiently. As a result, we observed our model performs great in SRL task. Moreover, our proposed module can be adopted as a complementary to any work utilizing the stacked self-attentions.

We present a comparative performance study using two benchmark data sets, CoNLL-2005 and CoNLL-2012, widely used in the literature. Our model outperforms the previous state-of-the-art approaches by 0.6, 1.1 and 0.2 F1 scores in WSJ test, Brown test and OntoNotes test, respectively. Experimental results also show that our model effectively captures the hierarchical information of long complex sentences. Moreover, we discuss that our architecture allows for more parallelization to compute the model efficiently.

The remaining of the paper is organized as follows: Section 2 describes the backgrounds of SRL and attention mechanisms. Section 3 presents our proposal deep network architecture. Section 4 shows the experimental study. Section 5 discusses related work. We conclude our work in Section 6.

2. Backgrounds

2.1. Semantic Role Labeling

SRL is the process of computational identification and classification of the arguments in given text into predefined labels. The label denotes the role of words with respect to predicates in a sentence. For example, given a sentence, "Some nights, he slept under his desk.", the result of SRL is presented as follows:

$$[_{\text{TMP}} \text{ Some nights,}], [_{\text{A0}} \text{ he}][_{\text{V}} \text{ slect}][_{\text{LOC}} \text{ under his desk.}]$$

where A0 represents the actor, V represents the predicate, TMP denotes the timing of the action, and LOC indicates the location of the actor.

SRL consists of three basic steps: (1) predicate identification, (2) argument assignment, and (3) classifying arguments. In the first step, syntactic analysis, Part-of-speech (POS) tagging, is performed

to detect clauses and frames. Based on the POS tags attached to tokens, a single main verb is selected as a predicate. In some problem settings, a predicate is given with a sentence to be parsed. In the second step, it chooses candidate arguments from non-argument tokens. It is common to build a parse tree and predicts possible dependent tokens to the selected predicate. Finally, any classification algorithms can be applied to find a possible solution for the sequence labeling problem. In recent studies, [7,9–11], they adopt end-to-end deep neural network models to take only sentences as inputs to labeling semantic roles for corresponding words.

2.2. Attention Mechanisms

Attention is a trainable alignment of an input and an output. In NLP, this mechanism is used to represent how much the input symbols are associated with generating a certain output symbol. The most salient symbols in the input are learned as *focused* by adjusting the weights during training. The mechanism is often formulated jointly with the sequence to sequence model that consists of two modules: an encoder and a decoder [16]. In the encoder, it takes the input sentence x, a sequence of vectors $x = (x_1, \ldots, x_{t_x})$. The output vector of the encoder is called *context vector* c that contains latent features h of the input x. The vector is commonly generated by RNN hidden states through time t such that

$$c = f(\{h_1, \ldots, h_t\}), \tag{1}$$

where $h_t \in \mathbb{R}^n$ is a hidden state at time t. f is a nonlinear function. The decoder is trained to generate the next token $y_{t'}$, given the vector c and all the previous predicted words $\{y_1, ..., y_{t'-1}\}$. The context c is used while training as

$$p(y'_t | y_1, ..., y_{t'-1}, x) = g(y_{t'-1}, s_{t'}, c_{t'}), \tag{2}$$

where g is a nonlinear function that outputs the probability of y'_t. s'_t is the hidden state computed by $s_i = l(s_{i-1}, y_{i-1}, c_i)$. $c_{t'}$ is the context vector computed at each decoding step t'. Without the attention mechanism, s_i is a function $f(s_{i-1}, y_{i-1})$, which removes the context vector. The context vector c_i is depends on a sequence of hidden states as

$$c_i = \sum_{j=1}^{t_x} \alpha_{ij} h_j. \tag{3}$$

where the weight α_{ij} of each state is computed by

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{t_x} exp(e_{ik})}, \tag{4}$$

where e_{ij} is a score a of association between the hidden state s_{i-1} of the decoder and the hidden state h_j of the encoder. The score a can be modeled based on the alignment table that represent a direct allocation, which can be jointly trained via end-to-end manner. In this work, we adopt several variants proposed in [14] to compute the score, which is denoted as *attention score*.

3. Selectively Connected Attentive Representation

To tackle the problems mentioned in the introduction, we propose a novel deep architecture for SRL. Inspired by the self-attention mechanism [14], our model train the correlation between words in a sentence based on the multiple layers of attentive representation. Upon the staked self-attention modules, we construct a connecting mechanism over the layers of attentive representations to capture the hierarchical structure in a sentence. In the following subsections, we describe the details of our proposed architecture comprising embedding modules, attentive representation modules, selective connection modules, and its training process.

3.1. Embedding

Word embedding is the first step of using neural networks to feed natural word symbols as processing inputs. To encode the natural word sequence into a set of vectors, the distributional word representation [17] is often used. First, each word in the training data is converted to the one-hot vector. Given a predicate, a predicate mask is similarly projected to the same one-hot vector space. Then, the (pre-trained) embedding matrix projects the two one-hot vectors to d-dimensional space. The projected vectors are concatenated to a vector ev_w, that is called *embedding vector*. As a pre-trained embedding matrix, we use Embeddings from Language Models (ELMo) [18] as in [11], which shows the state-of-the-art results. In our model, we set the size of the projection space as same as the size of the attentive representation described following subsections for simplicity. Because the self attentive representation ignores the positions of words, we should encode the positional information of words. To distinguish the different positions, we project the word positions to the same space of word embedding by using the well-defined function in the related work [14]. A sine function and cosine function are used for different frequencies:

$$\begin{aligned} ev_p(pos, 2i) &= sin(pos/10000^{(}2i/d)) \\ ev_p(pos, 2i+1) &= cos(pos/10000^{(}2i/d)), \end{aligned} \tag{5}$$

where pos denotes the word index and i denotes the number of dimensions.

To incorporate the sequential nature of language, we build an optional encoding layer based on the bidirectional LSTMs as in the state-of-the-art approach [11]. This layer exploits two parallel LSTMs processing opposite directions of processing input sequences. The two outputs of this layer are concatenated and applied max pooling to reduce the dimension to the embedding space. The output is fed to the first layer of attentive representation module, which will be described in the next subsection. The output this layer is formulated as follows:

$$\begin{aligned} \overrightarrow{h_t} &= \overrightarrow{LSTM}(x_t, \overrightarrow{h}_{t+1}) \\ \overleftarrow{h_t} &= \overleftarrow{LSTM}(x_t, \overleftarrow{h}_{t-1}) \\ ev_s &= \overrightarrow{h_t} + \overleftarrow{h_t}. \end{aligned} \tag{6}$$

The final embedding vector is represented as $L \times \{ev_w \odot ev_p \odot ev_s\}$ where L is the length of the sentence, and $\odot$ is the element-wise multiplication.

3.2. Attentive Representation

Basically, our model mainly relies on the self-attention mechanism [14] to capture the semantic roles of the words in a sentence. The self-attention has been successfully applied to draw the dependency structure within a single sequence. In recent studies [15,19], the self-attention is widely used because it is not only avoiding recurrent updates during training, but also capturing the intra-relationship in sentences. The most relevant work [15] to ours has achieved state-of-the-art performance (without pre-trained embedding models) in 2017 by parallelizing the encoding and decoding processes by using multi head self-attention. Inspired by the prior work, we build the dependency structure featured for SRL by densely connecting the every word parts to each other. Figure 1 illustrates the multi head self-attention module.

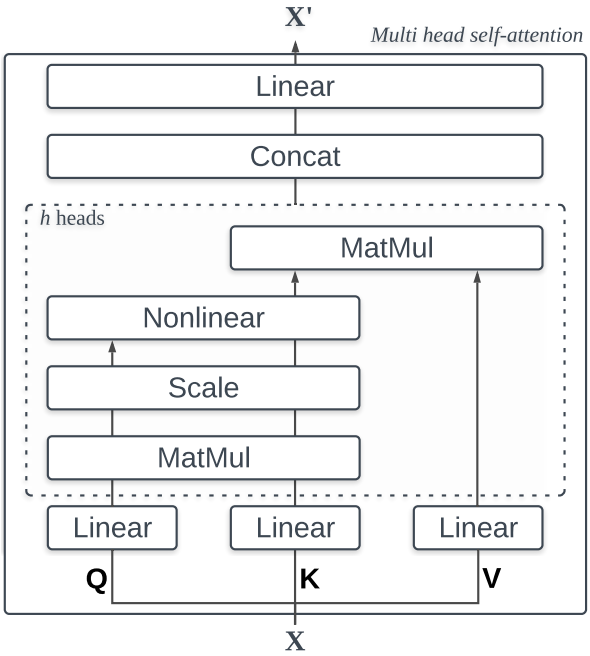

Figure 1. Multi head self-attention.

As a basis, the scaled dot-product attention is adopted to compute the attention score. The dotted block denoted as *h*-heads of self-attention depicts a single computation flow of the scaled dot-product attention. Also, two variants can be used for efficient computation: dot-product attention and additive attention. The dot-product attention utilizes matrix production, which allows faster computation [15]. Given a matrix of n query vectors $\mathbf{Q} \in \mathbb{R}^{n \times d}$, keys $\mathbf{K} \in \mathbb{R}^{n \times d}$ and values $\mathbf{V} \in \mathbb{R}^{n \times d}$ for $\mathbf{X}$, the scaled dot product attention computes the attention scores based on the following equation:

$$a(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}})\mathbf{V}, \tag{7}$$

where d is the number of hidden units of the network. Without $\sqrt{d}$, the score computes the dot-product attention. Additive function is just a simple concatenation.

For multiple heads of the self-attention, we adopt linear projections by $\mathbf{W}_i^Q \in \mathbb{R}^{n \times d/h}$, $\mathbf{W}_i^K \in \mathbb{R}^{n \times d/h}$, $\mathbf{W}_i^V \in \mathbb{R}^{n \times d/h}$, which correspond to queries, keys and values, respectively. The attention score a calculates the association between queries and keys for values to generate the attentive representation of the vectors X. As a result, our model is incorporating multiple parallel channels, that is, *h*-multi heads m_i, to represent the different learned semantics from the inputs:

$$m_i = a(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V), \tag{8}$$

To combine the multiple channels as a single representation, we concatenate them and apply linear projection via a feed forward layer as $X' = Concat(\mathbf{m}_1, ..., \mathbf{m}_h) \times \mathbf{W}_\mathbf{A}$. The Figure 2 illustrate the module called *attentive representation* that makes the multi head self-attention module to be trainable. Each module contains the multi head self-attention sub-module and fully connected sub-layer, followed by a sub-layer for normalization. Also, we adopt residual connections around each of the sub-layers described as blocks. The number h of the parallel layers is given as a hyper-parameter. To deepen the model, we stack the attentive representation modules by N sequential layers. It is worth noting that the depth of our model affects the performance on labeling accuracy, which means the learned semantic representation would be more robust according to the depth. By adopting the multiple heads and the stacking, our model can capture the correlative relationships within a sentence effectively.

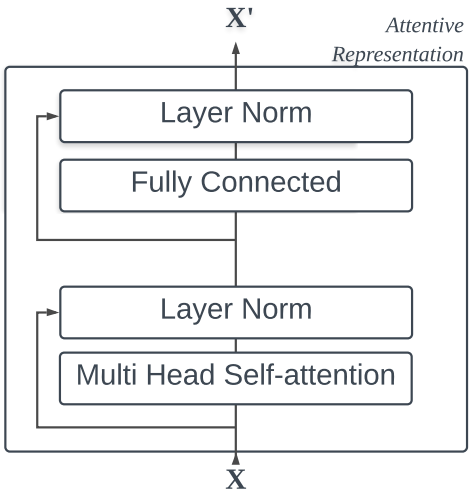

Figure 2. Attentive Representation.

3.3. Selective Connection

The stacked attentive representations is suitable for the pairwise connections, which is a densely populated structure, between words in a sentence. In other words, it can capture the associations among single symbols. However, it cannot learn the features related to the spatial constraints like hierarchy. Our model combines several components from previous studies [14,15,19]. However, our work is different from them in that we connect the identical layers of attentive representation to learn the spatial constraints in sentences. The connection is built with a gated unit that selectively updates the latent feature S_i over the variable-depth model, which comprises N identical multi head self-attention modules. We propose a novel module called *selective connection* that maintains the features capturing the spatial constraints, such as hierarchical structure within the sentence. This module adjusts the flow of information through the network, which is the orthogonal characteristic of self-attention mechanism. As a result, our model can manage where to focus over the sentence during encoding its representation. The motivation of this proposal is that the adaptation of the hierarchical representation learning from other fields [20,21], would be beneficial to SRL tasks.

The input representation X_i is generated from the i-th layer of attentive representation module. The initial state S_0 is set to be a vector whose elements sampled from a Gaussian distribution with mean 0 and variance $1/s$ where s denotes the size of the feature vector. According to each attentive representation module steps, the state S_i is updated as follows:

$$S_i = (1 - z_i)S_{i-1} + z_i\hat{S}_i, \tag{9}$$

where z is a selective modulator that decides how much the features generated from the attentive representation module activated. The modulator z is computed by

$$z_i = f(W_z X_i + U_z S_{i-1}), \tag{10}$$

where f is a nonlinear function, for example, sigmoid. W_z and U_z are weight matrices to sum the existing state and the new state to be selected. The candidate state feature $\hat{S}_i$ is computed by gating units g as follows:

$$\hat{S}_i = tanh(WX_i + U(g_i \odot S_{i-1}), \tag{11}$$

where $\odot$ is an element-wise multiplication. The gating unit g selectively chose the update z to be activated. The computation of the unit is performed as follows:

$$\hat{g}_i = f(W_g X_i + U_g S_{i-1}). \tag{12}$$

The Figure 3 illustrates the computational flow in the selective connection module.

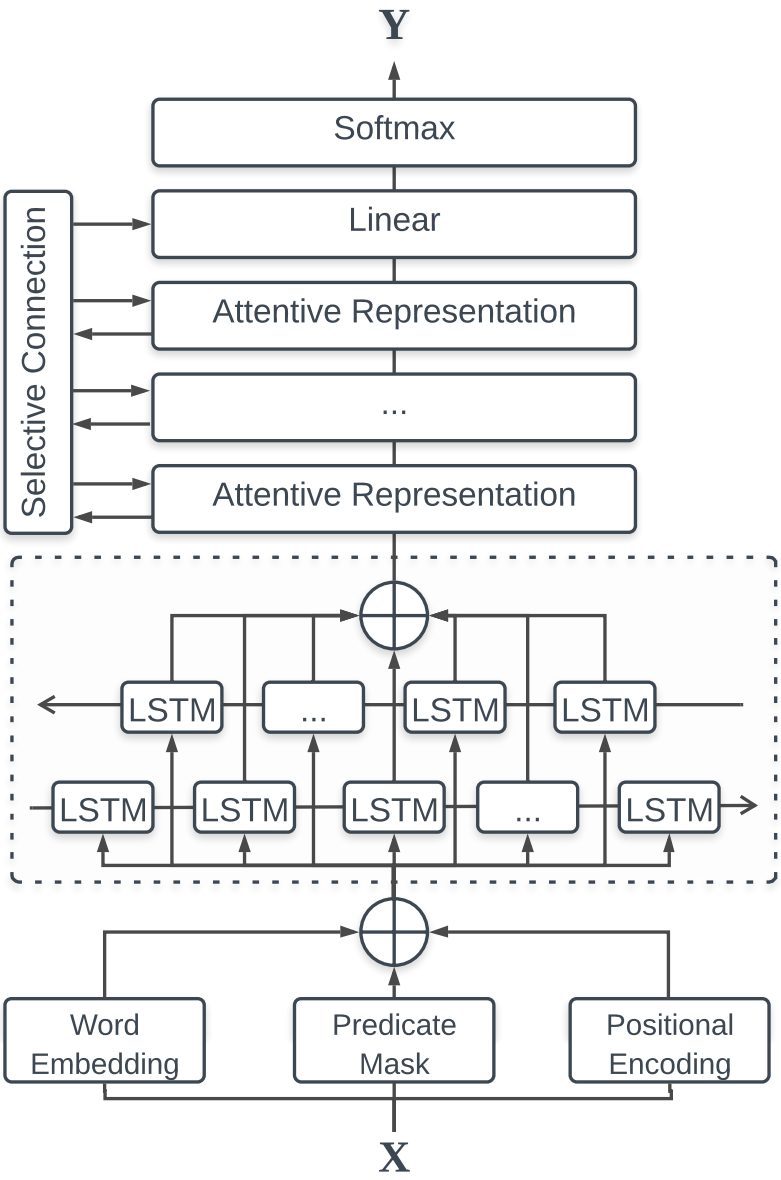

Figure 3. Selective Connection.

3.4. Overall Architecture

The architecture contains the attentive representation module, the selective connection module and embedding module described above. The Figure 4 illustrates the overview of proposed architecture to generate the final probabilities given input sequence.

Our model builds on the stacked encoders inspired by [14] for attentive representation of a sentence. The architecture composes of a stack of N identical attentive representation modules, and each module is updated using the selective connection module. We broaden the model by allowing parallel self-attention channels, namely, h-heads, to synthesize the latent information. The number h is a hyper-parameter which can be set. With the stacked connections and the multiple channels, the model can capture the word relationships regardless of their sequential steps without losing useful linguistic features such as, position and hierarchical structure. In the bottom layer, the embedding module to project the one-hot vectors of input sentence X to uniform space. Three sub-modules are adapted for sentence embedding: Word Embedding, Predicate Mask and Positional Encoding. The outputs of

the sub-modules are combined as the input to the optional module: the bidirectional LSTM encoders. The result of the encoder is fed to the first attentive representation module, which is the core layer to densely connect the identical input sequence features. Through the N repetition, the selective connection module, which is a simple, spatially aware gated network, learns the latent features among the attentive representations. As a result, the connection module maintains the hierarchical information captured through the N steps. To stabilize the activation of deep structure, we apply a residual connection followed by layer normalization. Every module is applied dropout as in [14], but we exclude them from the figures for clear presentation. At the top, we add a fully connected layer and a softmax layer to predict the probabilities of semantic role labels for the whole sentence.

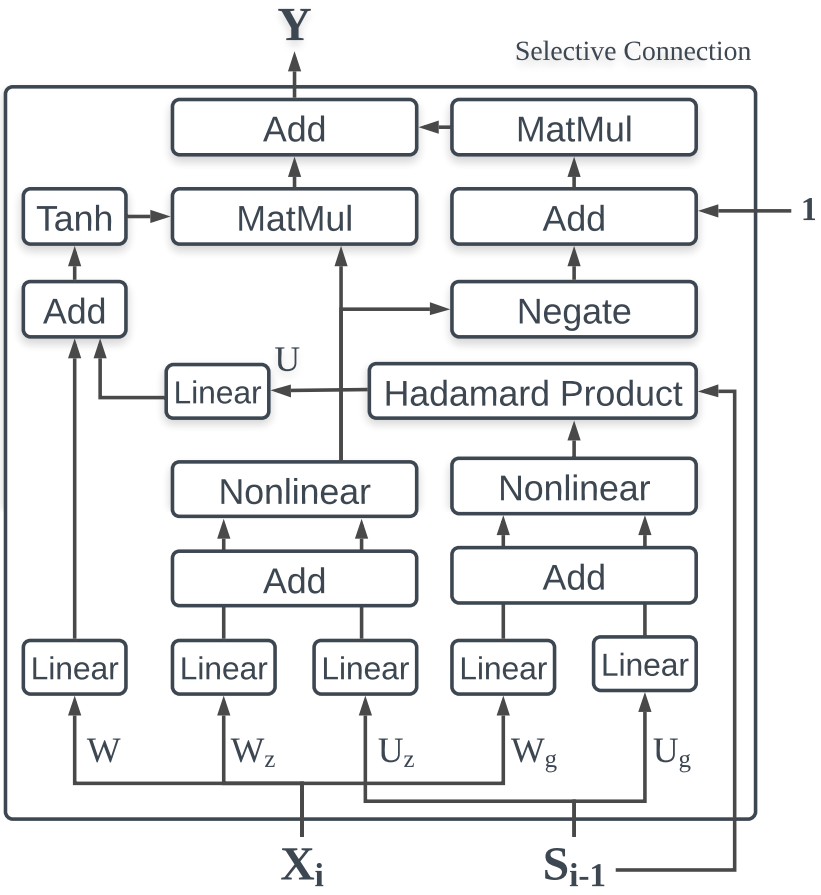

Figure 4. Overall architecture.

3.5. Training

The total process of training can be described as follows. Given an input sequence $\mathbf{x} = \{x_1, x_2, ..., x_n\}$, the log-likelihood of the corresponding label sequence $\mathbf{y} = \{y_1, y_2, ..., y_n\}$ is $\log p(\mathbf{y}|\mathbf{x}; \theta) = \sum_{t=1}^{N} \log p(y_t|\mathbf{x}; \theta)$. As a training sample x, a sequence of words w with predicate masks m and positional encoding result are combined to the network. Then, the next layer takes the densely populated connections, self attention, as the inputs to capture the dependency structure in them. The dense connection is updated via N steps of self-attention modules. At each step, the model selectively takes the previously generated features as an input to the next. At last, we take the outputs of the topmost self-attention layer are fed to softmax layer for the final prediction. We try to maximize the probabilities of the correct output labels with the input sequence over the entire training set given as follows:

$$\begin{aligned} \log p(y_t|\mathbf{x};\theta) &= p(y_t|\mathbf{h}_t;\theta) \\ &= softmax(\mathbf{W}\mathbf{h}_t)^\top \delta_{y_t}), \end{aligned} \tag{13}$$

where $\mathbf{W}$ is the softmax matrix and δ_{y_t} is Kronecker delta with a dimension for each output label. At the inference, we use the average pooling layer to combine all the outputs to get the label probability. To train the model, we use the multi-channel cross entropy loss. With the loss, every layer has the ability to learn the word information. Every layer can generate the current label.

4. Experiments

In this section, we report a comparative experimental study using two data sets: CoNLL-2005 shared task and the CoNLL-2012 shared task.

4.1. Datasets

We use the CoNLL-2005 dataset and the CoNLL-2012 dataset for our experiment to compare with previous work. The CoNLL-2005 has 44,020 sentences with POS tags, clauses with BIO tags, role labels for predicates and its arguments. This data consists of sections of the Wall Street Journal part of the Penn TreeBank. The test set includes three sections of the Brown corpus. The CoNLL-2012 dataset is extracted from the final version, v5.0 OntoNotes corpus. The detail setting for cross validation setting on training and test sets is described as in [22,23].

4.2. Experimental Setup

We initialize the weights of all sub-layers as random orthogonal matrices. For other parameters, we initialize them by sampling each element from a Gaussian distribution with mean 0 and variance $1/d$. The embedding matrix is initialized by the pre-trained ELMo model with fine-tuned parameters [18]. Words that habe not been appeared in the ELMo are changed to the special symbol <UNK>. We carefully discuss the impact of using pre-trained model in NLP tasks. We set the dimension of word embeddings and predicate mask is set to 512. The number of hidden units d to 100. The number of self-attention heads h is set to 16. To find the optimal structure, we vary the number of attentive representation layers from 4 to 10. The optional layer of bidirectional LSTMs (BiLSTMs) is initialized with random orthogonal matrices as in [24]. The dropout rate is set to be 0.8. The momentum and the weight-decay are 0.8 and 0.9 respectively. We use the AdaDelta optimizer with $\epsilon = 1 \times e^{-6}$ and $\rho = 0.95$ and mini-batch size is set to 80. Clipping gradient is done with threshold 1.0. The number of symbols in the batch is 4096 for the CoNLL-2005 dataset and 8192 for the CoNLL-2012 dataset. The learning rate is initialized to 0.01. We train our best model for 400 epochs, which takes about five days on two Titan Xp GPUs. We compare our model with the best single models of previous studies.

4.3. Experimental Results

We compare our models to five existing studies: two bidirectional LSTM-based models denoted as He et al. [9,11], include the state-of-the-art SRL model, and a self-attention-based model denoted as Tan et al. [15] and others [7,25]. Tables 1 and 2 shows our model performance with the reported results of the previous studies. We summarize the results of the best single models with the metrics in terms of precision (P), recall (R) and F1. Our proposed model achieves $F1$ scores of 86.6 and 77.2 on WSJ test and Brown test of CoNLL 2005 shared tasks, respectively. In CoNLL 2012 shared task, our model achieves best $F1$ scores of 83.6. In some senses, these may be small improvements on both datasets compared to the existing studies. However, we have observed several useful characteristics in the SRL task. For example, with ELMo pretrained model, $F1$ scores are increased from 83.3 to 86.6. Based on the study, it is worth noting that the pretrained embedding models with our architecture are important to general downstream tasks as well as sentence encoding. The key issue in SRL would be

incorporating the fine-tuning with the language models. The results explain our model is competitive to predict the semantic role labels compared to previous work.

Table 1. Experimental results on the CoNLL-2005.

Model	Development			WSJ Test			Brown Test		
	P	R	F1	P	R	F1	P	R	F1
Ours+ELMo	83.1	85.6	85.5	86.4	81.0	86.6	75.8	77.6	77.2
Ours	82.8	82.6	82.2	83.6	80.1	83.3	73.5	74.0	74.9
He et al. (2018)	-	-	85.3	84.8	87.2	86.0	73.9	78.4	76.1
Tan et al.	82.6	83.6	83.1	84.5	85.2	84.8	73.5	74.6	74.1
He et al. (2017)	81.6	81.6	81.6	83.1	83.0	83.1	72.9	71.4	72.1
Zhou and Xu	79.7	79.4	79.6	82.9	82.8	82.8	70.7	68.2	69.4
FitzGerald et al.	81.2	76.7	78.9	82.5	78.2	80.3	74.5	70.0	72.2

Table 2. Experimental results on the CoNLL-2012.

Model	Development			OntoNotes Test		
	P	R	F1	P	R	F1
Ours+ELMo	82.9	81.6	81.9	83.2	83.8	83.6
Ours	81.8	81.2	81.3	81.4	82.4	82.8
He et al. (2018)	81.8	81.4	81.5	81.7	83.3	83.4
Tan et al.	82.2	83.6	82.9	81.9	83.6	82.7
He et al. (2017)	81.7	81.4	81.5	81.8	81.6	81.7
Zhou and Xu	-	-	81.1	-	-	81.3
FitzGerald et al.	81.0	78.5	78.9	81.2	79.0	80.1

To better understand the effect of the proposed modules, we compare the performance of using the selective connection in self-attention based architectures. The Figure 5 shows the model performance, according to the different numbers of the depth of the model. With increasing number of model depths, $F1$ values of both models are gradually increased until it has 9 layers of attentive representation modules. The training times in the Figure 5 are normalized by setting the medians to be zero to present the improvement of efficiency. Training time difference due to the selective connection is relatively small from 6 to 10 layers, which is within the region of higher performance. Although the hyperparameters we used are slightly different from the work [15], the tendencies of training times increase are similar. For more than 4 layers of multi head self-attentions, our model with the selective connections performs much better than others. This means that the adoption of selective connection enhances the training efficiency.

For parallelism, we compare the time to complete varying the number of epochs given GPU settings. By assigning a model to 2 GPUs, we have observed that the training time is reduced by 0.62 compared to the single GPU model. For data parallelization, the training time is reduced by 0.67. Figure 6 shows how the different parallelization method competes with single models.

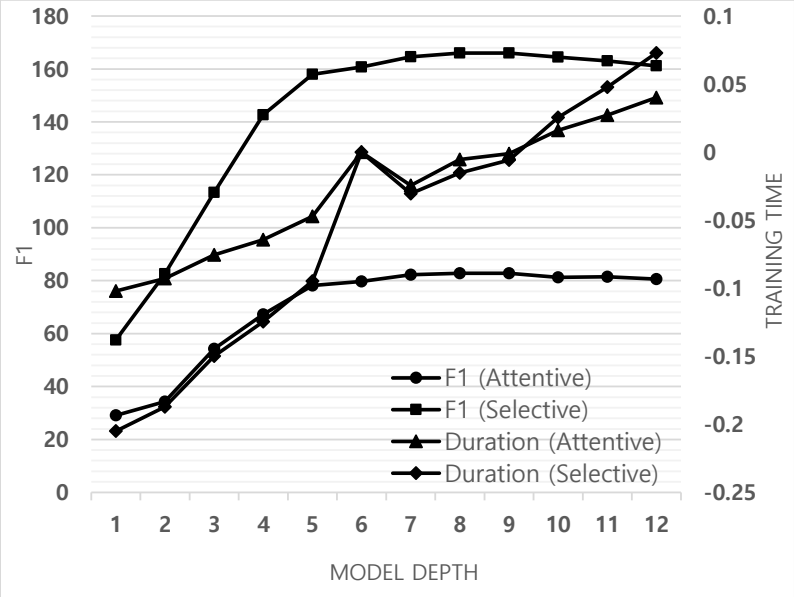

Figure 5. Performance changes with respect to model depth.

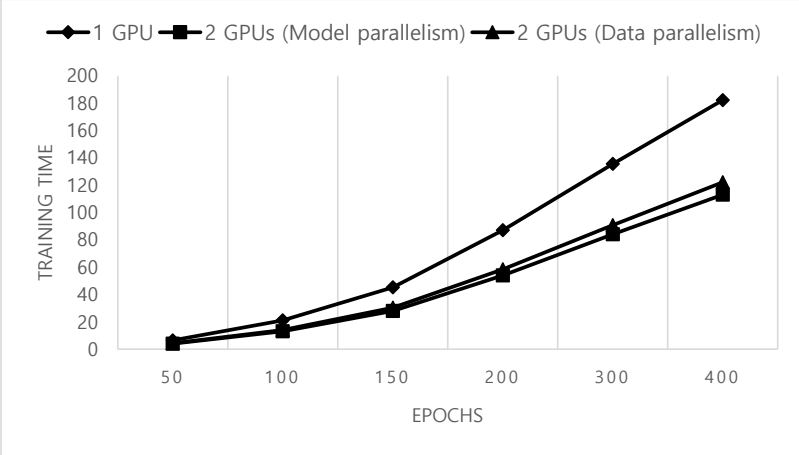

Figure 6. Training time difference for parallelization.

Table 3 summarizes *F*1 values of our model ablations on the CoNLL 2005 development set. We ablate our full model by removing, (1) the BiLSTM layers (denoted as Seq), (2) the selective connection module (denoted as SC) and (3) the ELMo model (denoted as Emb). Without the selective connection, the stack of attentive representation is constructed only with the fully connected layer followed by layer normalization. As the increasing number of layers, the model achieves higher *F*1 scores until it has 8 layers. We report the ablation result with the increasing numbers 4, 8 and 12 layers of the attentive representations to minimize the effect of network depth. Without the BiLSTM layers, the model achieves, at most, 79.7 *F*1, which means that the sequential nature in language should be encoded as an input as well as the positional encoding. Without the selective connection module, the performance drops by 7.1 in worst case. With Word2Vec model, *F*1 values cannot be reached to 80, which is the score of the early study [7]. All the performance results show the decreases with each ablation model. For parallelization, we have observed the training time decreased with the increasing number of GPUs used. Due to the limited GPU environment conducted, we cannot present the result in detail. We note that our model reduce the training time by 62 percentage from the baseline. As a future work, we plan to investigate the holistic effects of parallelization in training the models.

Table 3. Ablation study results.

	Emb	Seq	Depth	SC	F1
1	ELMo	BiLSTMs	4	Use	72.5
	ELMo	BiLSTMs	8	Use	85.2
	ELMo	BiLSTMs	12	Use	84.8
2	Word2Vec	BiLSTMs	4	Use	66.0
	Word2Vec	BiLSTMs	8	Use	83.1
	Word2Vec	BiLSTMs	12	Use	81.7
3	ELMo	FC	4	Use	64.2
	ELMo	FC	8	Use	79.7
	ELMo	FC	12	Use	75.3
4	ELMo	FC	4	N/A	71.3
	ELMo	FC	8	N/A	82.2
	ELMo	FC	12	N/A	82.2
5	Word2Vec	FC	4	N/A	68.9
	Word2Vec	FC	8	N/A	80.7
	Word2Vec	FC	12	N/A	80.3

To investigate the relationship between the selective connection and the SRL accuracy improvements, we qualitatively analyze the generated labels with phrases at each layer. Specifically, for each predicted label in the development set, we collect 7 predicted labels with their corresponding spans from the training set. Table 4 shows the inference results with given the predicate word "chased". We have observed the higher levels of attentive representation learn the higher syntactic structure captured by varying the depth of the network. At the ninth layer, the label for *accepted-from* [A1] is assigned to the phrase "the bird that saw the cat". This means the our model predicts the role labels for the span of words, which contains the new acceptor "the bird" and the new predicate "saw". The span could be understood by introducing a new sentence with label [A0] and [V] as specified in the lower layer 8. Our results show that our model understand the hierarchical structure over the compounding sentences consisting of acceptors to be accepted.

Table 4. Example sentence with parsed hierarchical information.

N-th Layer	Inferred Labels
4	[$_{A0}$rat] [$_{V}$ chased]
5	[$_{A0}$ the rat] [$_{V}$ chased] [$_{A1}$ the birds]
6	[$_{A0}$ the rat] [$_{V}$ chased] [$_{A1}$ the birds]
7	[$_{A0}$ the rat] [$_{V}$ chased] [$_{A1}$ the bird] [$_{A1}$ that saw the cat]
8	[$_{A0}$ the rat] [$_{V}$ chased] [$_{A1}$ the bird] that [$_{V}$ saw] [$_{A1}$ the cat]
9	[$_{A0}$ the rat] [$_{V}$ chased] [$_{A1}$ the bird that saw the cat] that [$_{V}$ ate] [$_{A1}$ the starling]
10	[$_{A0}$ the rat] [$_{V}$ chased] [$_{A0}$ the bird] that [$_{V}$ saw] [$_{A1}$ the cat] that [$_{V}$ ate] [$_{A1}$ the starling]

5. Related Work

Conventional SRL approaches [26–28] exploits the syntactic features for capturing the sentence structure. Early works [29,30] that introduce deep neural networks to SRL have tried to reduce the feature engineering. They have built some layers, such as convolutions and lexical embeddings, on the popular deep architectures. The pioneering work [7] having a deep network of eight levels of LSTMs outperforms the previous studies using syntactic information. Several works [10,31] also proposed a bidirectional LSTM-based model. Inspired by [7], the work [9] introduces an end-to-end SRL task as scoring BIO tags for a given word sequence. The model adopts two well-known structures, highway connections [32] to resolve the gradient vanishing problem. Also, A* search algorithm is used to determine promising BIO tags in a constrained way at a decoding stage. They have discussed the benefit of incorporating the explicit injection of labelled syntactic structures. Nevertheless,

they above methods may fail to train the model in several days because the training duration is too long. To facilitate parallelism, self-attention mechanism is adopted in recent work [15,19]. Moreover, the mechanism is well-known as a semantic representation for a sentence in many NLP tasks, such as machine reading comprehension [33]. The most recent work [11] has introduced an end-to-end network that jointly learn the predicates and arguments without any syntactic information. It generalizes the prior work [9] by using pretrained word embedding and BiLSTMs. To represent and learning dependency between predicates and arguments in the sentence, PathLSTM [34] are proposed. The idea is to learn embeddings of lexicalized dependency paths between predicate and each argument. They argued that syntactic information could improve the performance of SRL. They pointed out dependency is necessary for SRL and shows better performance with it. The work [35] introduced Random-walk based graph learning network. Many recent studies [10,19,34] have employed features extracted by the syntactic parser at the early stage to improve accuracy, we do not consider them in this paper because leveraging external knowledge is beyond our scope.

Discussion: As suggested by the many studies above, deep learning models are beneficial to SRL. In our study, we propose simple represenational model using self-attention mechanism. Studies on self-attention in the preivious studies often do not address the concerns about exploiting spatial information in the sentences. Complementary to those studies, we focus on adopting both on sequential and hierarchical structures.

The self-attention mechanism has been introduced [14], and have been popularly used in many NLP tasks: machine reading comprehension [33], natural language inference [36], text summarization [37], language understanding [38], image captioning [39] and machine translation [14], which yields the state-of-the-art results.

Discussion: Our model is developed based on self-attention mechanism, which is similar to the several recent studies [15,19]. However, our proposed module, selective connection, is different from the stacked multi head self-attention modules connected by the feed-forward layers. The module provides sparse connections to allow training complex structure in the context vectors.

6. Conclusions

We proposed a deep self-attention network for the task of SRL. A novel idea of selective connection is applied through the deep network of attentive representations to train the spatial constraints over sentences. Our model outperforms the state-of-the-art on two benchmark data sets, the CoNLL-2005 and the CoNLL-2012 shared task datasets. Our experimental results indicate that our model is competitive to train the spatial constraints in sentence for efficient SRL. Qualitative analysis explains the effect of learning hierarchical structure is beneficial. Future work will explore the training efficiency, maximizing the impact of parallelization. Also, we try to develop better training techniques and adopt recently released pre-trained word embedding models, the Bidirectional Encoder Representations from Transformers (BERT) [38].

Funding: Incheon National University (International Cooperative) Research Grant in 2018.

Acknowledgments: This work was supported by Incheon National University (International Cooperative) Research Grant in 2018.

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

References

1. Dowty, D. Thematic proto-roles and argument selection. *Language* **1991**, *67*, 547–619. [CrossRef]
2. Fillmore, C.J. Frame semantics and the nature of language. *Ann. N. Y. Acad. Sci.* **1976**, *280*, 20–32. [CrossRef]
3. He, L.; Lewis, M.; Zettlemoyer, L. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 643–653.

4. Liu, D.; Gildea, D. Semantic role features for machine translation. In Proceedings of the 23rd International Conference on Computational Linguistics, Beijing, China, 23–27 August 2010; pp. 716–724.
5. Collobert, R.; Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th International Conference On Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 160–167.
6. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [CrossRef]
7. Zhou, J.; Xu, W. End-to-end learning of semantic role labeling using recurrent neural networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, Beijing, China, 26–31 July 2015; pp. 1127–1137.
8. Cho, K.; Van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv* **2014**, arXiv:1409.1259.
9. He, L.; Lee, K.; Lewis, M.; Zettlemoyer, L. Deep Semantic Role Labeling: What Works and What's Next. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 473–483.
10. Marcheggiani, D.; Frolov, A.; Titov, I. A Simple and Accurate Syntax-Agnostic Neural Model for Dependency-based Semantic Role Labeling. In Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, BC, Canada, 3–4 August 2017; pp. 411-420
11. He, L.; Lee, K.; Levy, O.; Zettlemoyer, L. Jointly predicting predicates and arguments in neural semantic role labeling. *arXiv* **2018**, arXiv:1805.04787 .
12. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2014**, arXiv:1409.0473.
13. Lin, Z.; Feng, M.; Santos, C.N.d.; Yu, M.; Xiang, B.; Zhou, B.; Bengio, Y. A structured self-attentive sentence embedding. *arXiv* **2017**, arXiv:1703.03130.
14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
15. Tan, Z.; Wang, M.; Xie, J.; Chen, Y.; Shi, X. Deep Semantic Role Labeling with Self-Attention. *arXiv* **2017**, arXiv:1712.01586.
16. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*; Montreal, QC, Canada, 8–13 December 2014; pp. 3104–3112.
17. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 1188–1196.
18. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. *arXiv* **2018**, arXiv:1802.05365.
19. Strubell, E.; Verga, P.; Andor, D.; Weiss, D.; McCallum, A. Linguistically-Informed Self-Attention for Semantic Role Labeling. *arXiv* **2018**, arXiv:1804.08199.
20. Socher, R.; Lin, C.C.; Manning, C.; Ng, A.Y. Parsing natural scenes and natural language with recursive neural networks. In Proceedings of the 28th International Conference on Machine Learning (ICML-11), Bellevue, WA, USA, 28 June–2 July 2011; pp. 129–136.
21. Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; Hovy, E. Hierarchical attention networks for document classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 1480–1489.
22. Carreras, X.; Màrquez, L. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005), Ann Arbor, MI, USA, 29–30 June 2005; pp. 152–164.
23. Pradhan, S.; Moschitti, A.; Xue, N.; Uryupina, O.; Zhang, Y. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In Proceedings of the Joint Conference on EMNLP and CoNLL-Shared Task, Association for Computational Linguistics, Jeju Island, Korea, 13 July 2012; pp. 1–40.
24. Saxe, A.M.; McClelland, J.L.; Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv* **2013**, arXiv:1312.6120.

25. FitzGerald, N.; Täckström, O.; Ganchev, K.; Das, D. Semantic Role Labeling with Neural Network Factors. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP '15), Lisbon, Portugal, 17–21 September 2015; pp. 960–970
26. Pradhan, S.; Ward, W.; Hacioglu, K.; Martin, J.H.; Jurafsky, D. Semantic role labeling using different syntactic views. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Ann Arbor, MI, USA, 25–30 June 2005; pp. 581–588.
27. Surdeanu, M.; Turmo, J. Semantic role labeling using complete syntactic analysis. In Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005), Ann Arbor, MI, USA, 29–30 June 2005; pp. 221–224.
28. Palmer, M.; Gildea, D.; Xue, N. Semantic role labeling. *Synth. Lect. Hum. Lang. Technol.* **2010**, *3*, 1–103. [CrossRef]
29. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P.P. Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
30. Fonseca, E.R.; Rosa, J.L.G. A two-step convolutional neural network approach for semantic role labeling. In Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA, 4–9 August 2013; pp. 1–7.
31. Marcheggiani, D.; Titov, I. Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. *arXiv* **2017**, arXiv:1703.04826.
32. Srivastava, R.K.; Greff, K.; Schmidhuber, J. Training very deep networks. In Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 2377–2385.
33. Cheng, J.; Dong, L.; Lapata, M. Long short-term memory-networks for machine reading. *arXiv* **2016**, arXiv:1601.06733.
34. Roth, M.; Lapata, M. Neural Semantic Role Labeling with Dependency Path Embeddings. *arXiv* **2016**, arXiv:1605.07515.
35. Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference On Knowledge Discovery And Data Mining, New York City, NY, USA, 24–27 August 2014; pp. 701–710.
36. Parikh, A.P.; Täckström, O.; Das, D.; Uszkoreit, J. A decomposable attention model for natural language inference. *arXiv* **2016**, arXiv:1606.01933 .
37. Paulus, R.; Xiong, C.; Socher, R. A deep reinforced model for abstractive summarization. *arXiv* **2017**, arXiv:1705.04304.
38. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
39. Zhou, L.; Zhou, Y.; Corso, J.J.; Socher, R.; Xiong, C. End-to-end dense video captioning with masked transformer. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8739–8748.