

Article

Not peer-reviewed version

---

# Empowering Sentence Representation with Semantic-Enhanced Tree-LSTMs

---

Andrew H. Chris , Fei Lee , [Woods Ali](#) \*

Posted Date: 28 November 2023

doi: 10.20944/preprints202311.1766.v1

Keywords: Natural Language Processing; Semantic Parsing; &nbsp; Syntactic Dependencies; &nbsp; Language Understanding



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

# Empowering Sentence Representation with Semantic-Enhanced Tree-LSTMs

Andrew H. Chris<sup>1</sup>, Fei Lee<sup>2</sup> and Woods Ali<sup>1,\*</sup>

<sup>1</sup> University of Charleston

<sup>2</sup> National University of Singapore

\* Correspondence: aliwoods12345@gmail.com

**Abstract:** The Semantic-Enhanced Tree-LSTM (SeT-LSTM) network, a novel advancement in the realm of linguistic modeling, marks a significant step forward from traditional Tree-based Long Short Term Memory (LSTM) networks. By intricately weaving in the nuances of typed grammatical dependencies, SeT-LSTMs offer a more nuanced understanding of language semantics. Traditional models often overlook the semantic shift caused by variations in word or phrase roles, a gap this paper aims to fill by focusing on the types of grammatical connections, or typed dependencies, within sentences. Our proposed architecture, dubbed the *Semantic Relationship-Guided LSTM (SRG-LSTM)*, leverages a control mechanism to model the interplay between sequence elements. Additionally, we present a novel Tree-LSTM variant, the *Semantic Dependency Tree-LSTM (SDT-LSTM)*, which integrates dependency parse structures with dependency types for more robust sentence embedding. The SDT-LSTM demonstrates superior performance in Semantic Relatedness Scoring and Sentiment Analysis compared to its predecessors. Qualitatively, it shows resilience to changes in sentence voice and heightened sensitivity to nominal alterations, aligning well with human intuition. This research underlines the pivotal role of grammatical relationships in sentence understanding and paves the way for further exploration in this domain.

**Keywords:** natural language processing; semantic parsing; syntactic dependencies; language understanding

## 1. Introduction

Natural Language Processing (NLP) encompasses a broad array of tasks, with sentence modeling standing as a foundational element in disciplines such as Sentence Classification [1], Paraphrase Identification [2,3], Question Answering [4,5], Sentiment Analysis [6,7], and Semantic Similarity Scoring [8,9]. The crux of these tasks lies in the effective representation of word meanings, typically achieved through sophisticated neural embedding techniques [10]. These embeddings serve as the building blocks for deriving sentence semantics, traditionally approached through either Bag-of-Words (BOW) [11] or sequential models [7,12]. While the BOW model simplifies sentences to mere word collections, employing basic vector operations for composition [11], sequential models view text as a linear progression of words, leveraging deep learning techniques like Long Short Term Memory (LSTM) networks [13] and Convolutional Neural Networks (CNN) [6,14] for semantic matching tasks.

Despite their widespread usage, these traditional models often struggle to capture the complex, non-linear dependencies prevalent in natural languages. This is a critical shortcoming, as the nuanced interplay of words in a sentence often holds the key to its true meaning. To address this gap, tree-structured models have been developed, utilizing architectures that mirror the parse trees of sentences [7,8,15–19] or employing latent trees designed for specific linguistic tasks [20,21]. Among these, Tree-RNNs, particularly those grounded in grammatical structures, offer a more sophisticated approach, organizing neural network units along the nodes of either a binarized constituency parse tree (CT) or a dependency parse tree (DT). The distinction between CT-RNN and DT-RNN architectures lies primarily in their approach to compositional parameters [7]. The CT-RNN model places word embeddings exclusively at leaf nodes, with parent nodes emerging from the combination of left and

right child nodes. Conversely, in the DT-RNN setup, each node is associated with a word embedding and receives an unordered list of child nodes as inputs. Empirical studies by [7] have shown that DT-LSTM models excel in tasks like semantic relatedness scoring, while CT-LSTM models are better suited for sentiment classification.

However, a significant limitation of these tree-based models is their oversight of the type of word relationships, a factor that plays a vital role in the overall semantic understanding of a sentence. Words in a sentence can form various grammatical relationships, reflected in the sentence's dependency tree structure. These relationships, known as Typed Dependencies [27] (encompassing both Stanford typed dependencies and Universal Dependencies), are instrumental in contributing to the sentence's semantic fabric.

To illustrate the importance of these typed dependencies, consider the sentences: 1) "Dogs chased cats in the garden" and 2) "Cats chased dogs in the garden." In both instances, the word-pair (Dogs, chased) maintains a parent-child relationship within their respective dependency trees. However, the typed dependency in the first sentence is "dobj" (direct object), indicating "Dogs" as the object of the action "chased", while in the second sentence, it is "nsubj" (nominal subject), signifying "Dogs" as the subject executing the action "chased". This subtle difference in grammatical relations is pivotal in shaping the overall meaning of the sentences. Yet, models that do not consider these distinctions fail to accurately capture the nuanced semantics of such sentences.

The implications of typed dependencies extend beyond semantic interpretation to sentiment analysis. Sentiment analysis aims to ascertain the sentiment polarity of a text, typically categorizing it as positive, negative, or neutral. For instance, phrases like "white blood cells destroying an infection" and "an infection destroying white blood cells" [19],

despite sharing the same words and tree structure, convey starkly different sentiments. The former is imbued with a positive connotation, while the latter carries a negative sentiment. Models relying solely on bag-of-words techniques or parse structures devoid of dependency type considerations are incapable of distinguishing between these semantic and sentiment nuances. This highlights the necessity for a deep learning model that is not only aware of but also actively incorporates typed dependencies.

Recent research efforts [28–31] have attempted to address the limitations of classic deep learning models in handling syntactic categories by incorporating POS and constituency tags. However, the exploration into the role of typed dependencies remains relatively uncharted. A notable exception is the work of [32] with their Semantic Dependency Tree-RNN model (SDT-RNN), a recursive neural network formulated based on the dependency tree. The SDT-RNN model distinguishes itself by training separate feed-forward neural networks for each dependency type, a method that, while promising, presents considerable data intensity and training complexity challenges.

In the realm of Dependency-based Long Short-Term Memory (D-LSTM) networks, [31] introduced an  $\alpha$ -weighted supporting component, derived from the subject, object, and predicate of the sentence, to augment the basic sentence representation generated by standard LSTM. Similarly, the Part-of-Speech based LSTM (pos-LSTM) model by [42] calculates the supporting component from the hidden representation of constituent words and their tag-specific weights. Intriguingly, the pos-LSTM model achieved optimal results when it utilized only the hidden representation of nouns in the sentence to compute the supporting component, underscoring the predominant role of noun words in sentence semantics. Both the D-LSTM and pos-LSTM models, while making strides in their respective domains, are not inherently syntax-aware, as they follow a sequential approach and employ grammatical structures solely for identifying semantic roles or POS tags. Despite this, their improvements over the baseline Manhattan LSTM (MaLSTM) model [12] are noteworthy.

Building on these insights, [8] proposed the Tag-Guided Hyper Tree-LSTM (TG-HTreeLSTM) model, which consists of a primary Tree-LSTM complemented by a hyper Tree-LSTM network. The hyper Tree-LSTM network utilizes a hypernetwork to dynamically predict the parameters of the main Tree-LSTM, leveraging POS tag information at each node of the constituency parse tree. Similarly, the

Structure-Aware Tag Augmented Tree-LSTM (SATA TreeLSTM) [28] employs an additional tag-level Tree-LSTM to supply auxiliary syntactic information to its word-level Tree-LSTM. These models have demonstrated significant improvements over their tag-unaware counterparts in sentiment analysis tasks, reinforcing the potential benefits of integrating syntactic information into deep learning models.

Motivated by these developments, our research aims to delve deeper into the role of grammatical relationships, particularly typed dependencies, in the semantic composition of language. This endeavor is guided by two primary objectives: 1) To conceptualize and introduce a versatile LSTM architecture capable of discerning and utilizing the type of relationships between elements in a sequence. 2) To develop an advanced deep neural network model, the Semantic Dependency Tree-LSTM (SDT-LSTM), optimized for learning a more nuanced semantic representation of sentences by harnessing both the dependency parse structure and the intricacies of typed dependencies between words.

In pursuit of these objectives, we have innovated an additional neural network module, the "Relation Gate," and integrated it into the LSTM architecture. This module serves as a regulatory mechanism, modulating the information flow between LSTM units based on an additional control parameter, termed 'relation'. Leveraging this "Relation-Gated LSTM" (R-LSTM), we have crafted the SDT-LSTM model, a sophisticated approach for computing sentence representations. This model, drawing inspiration from [7]'s Dependency Tree-LSTM, has demonstrated superior performance in two key sub-tasks: Semantic Relatedness Scoring and Sentiment Analysis.

The contributions of this paper are threefold:

1. The introduction of the Relation-Gated LSTM (R-LSTM) architecture, a novel framework that incorporates an additional control input to regulate the LSTM hidden state based on the type of relationship between words.
2. The development of the Semantic Dependency Tree-LSTM (SDT-LSTM) model, which utilizes R-LSTMs for learning sentence semantic representations over dependency parse trees, with a specific focus on typed dependencies.
3. An in-depth qualitative analysis that sheds light on the pivotal role of typed dependencies in enhancing language understanding, particularly in the context of semantic relatedness and sentiment analysis.

The subsequent sections of this paper are organized in the following manner: Section 2 delves into a comprehensive exploration of Long Short-Term Memory (LSTM) networks and elaborates on the nuances of the Dependency Tree-LSTM architecture. In Section 3, we meticulously detail the construction and underlying principles of the Semantic Relationship-Guided LSTM (SRG-LSTM), along with a thorough description of the Semantic Dependency Tree-LSTM (SDT-LSTM) model's framework. Section 4 is dedicated to elucidating our experimental methodology and presenting the results obtained, effectively demonstrating the superior performance of our newly proposed models. The paper reaches its culmination in Section 5, where we reflect on the insights gained from our research and propose potential avenues for future investigations in the realm of information extraction through the lens of syntactic dependency features.

## 2. Preliminary

### 2.1. Enhanced Long Short Term Memory (LSTM) Overview

Sequential neural architectures, particularly Recurrent Neural Networks (RNNs), offer a dynamic approach to processing input sequences of indeterminate lengths. In RNNs, each sequence element undergoes a transformation via a recurrent function. An RNN unit, at any given time step  $t$ , considers both the immediate input  $x_t$  and the preceding hidden state  $h_{t-1}$ . Depending on the specific application, the RNN's output could manifest as the final hidden state's output or as a series of outputs derived from each state.

Long Short Term Memory networks (LSTMs), a specialized RNN variant, excel in preserving information over extended sequences through their unique cell memory mechanism. LSTMs utilize a

trio of gates - the forget gate, input gate, and output gate, in tandem with dual memory states - cell state and hidden state, all represented as vectors in the  $d$ -dimensional real space,  $\mathbb{R}^d$ . Here,  $d$  is a critical hyperparameter defining the memory state dimension. The operations within an LSTM unit at any time step  $t$  are encapsulated in Equation 1.

$$\begin{aligned}
 i_t &= \sigma(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}), \\
 f_t &= \sigma(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}), \\
 o_t &= \sigma(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}), \\
 u_t &= \tanh(W^{(u)}x_t + U^{(u)}h_{t-1} + b^{(u)}), \\
 c_t &= i_t \odot u_t + f_t \odot c_{t-1}, \\
 h_t &= o_t \odot \tanh(c_t)
 \end{aligned} \tag{1}$$

The logistic sigmoid function  $\sigma$  and pointwise multiplication  $\odot$  are fundamental to the gating mechanisms. Each gating vector ( $f_t, i_t, o_t$ ) is generated by feedforward networks, leveraging both the current input  $x_t$  and the previous hidden state  $h_{t-1}$ . The LSTM's capacity to selectively filter and propagate critical information through its network hinges on these gates. Traditional LSTMs, with their inherently sequential nature, are not designed to recognize complex, non-linear input dependencies.

## 2.2. Syntactic Dependency Tree-LSTMs (SDT-LSTM)

In the realm of text processing, hierarchical structures within sentences, as depicted by parse trees, align naturally with tree-structured models. A notable advancement in this domain is the introduction of Tree-LSTMs by [7]. This concept involves LSTM units interconnected in a tree structure, allowing for multiple input sources per unit. [7] further delineated two Tree-LSTM variants: Child-Sum Tree-LSTMs and N-ary Tree-LSTMs, each optimized for specific tree structures in sentence representation. The former is designed for trees with nodes having an indeterminate number of unordered children, whereas the latter accommodates trees with nodes bearing up to N children in a fixed sequence. A Dependency Tree-LSTM (DT-LSTM), or in our revised terminology, a Syntactic Dependency Tree-LSTM (SDT-LSTM), is a child-sum Tree-LSTM applied to dependency parse trees. Each SDT-LSTM node processes an input vector  $x_t$  and multiple hidden states from its child nodes. Equation 2 details the transition dynamics of an SDT-LSTM unit at time step  $t$ .

$$\begin{aligned}
 h_{C(t)} &= \sum_{k \in C(t)} h_k, \\
 i_t &= \sigma(W^{(i)}x_t + U^{(i)}h_{C(t)} + b^{(i)}), \\
 f_{tk} &= \sigma(W^{(f)}x_t + U^{(f)}h_k + b^{(f)}), \quad k \in C(t) \\
 o_t &= \sigma(W^{(o)}x_t + U^{(o)}h_{C(t)} + b^{(o)}), \\
 u_t &= \tanh(W^{(u)}x_t + U^{(u)}h_{C(t)} + b^{(u)}), \\
 c_t &= i_t \odot u_t + \sum_{k \in C(t)} f_{tk} \odot c_k, \\
 h_t &= o_t \odot \tanh(c_t)
 \end{aligned} \tag{2}$$

where  $C(t)$  is the set of all child nodes of the node  $t$ .

In SDT-LSTM, unlike standard LSTMs, the gating vectors and cell states are influenced by multiple child units' hidden states. This model features multiple forget gates, one for each child unit, allowing selective emphasis on information from specific child nodes based on the task at hand. At every node  $t$ , the input vector  $x_t$  correlates to the headword's embedding at that node, while the hidden state  $h_t$  encapsulates the sub-tree's abstract representation rooted at  $t$ . The cell state of the parent node is derived from the aggregated cell states of its child nodes, leading to the

nomenclature "child-sum" LSTMs. Despite their ability to acknowledge non-linear word dependencies within sentences, neither Tree-LSTMs nor their variants explicitly leverage the nature of these syntactic dependencies. Our proposed approach, Semantic Relation-Gated LSTM (SRG-LSTM) and Syntactic Dependency Tree-LSTM (SDT-LSTM), introduces an additional control input  $z_t$  to the standard LSTM framework. This addition, a relation gate  $r_t$ , empowers the LSTM to modulate its hidden state  $h_t$  based on the dependency type, encoded in  $z_t$ . The detailed architecture and functionality of SRG-LSTM and SDT-LSTM are discussed in following section.

### 3. Our Method: Semantic-Enhanced Tree-LSTMs

#### 3.1. Syntactic Relation Gated LSTM Architecture (SRG-LSTM)

The Syntactic Relation Gated LSTM (SRG-LSTM) extends the capabilities of the standard LSTM architecture.

SRG-LSTM comprises four gates: the forget gate, input gate, output gate, and a novel relation gate, along with two memory states: the hidden state and the cell state. The operation of the relation gate and hidden state in SRG-LSTM at time step  $t$  with successor  $t'$  is described by equation 3. For a sequential model,  $t' = t + 1$ , while for tree-based models,  $t'$  is determined by the underlying tree topology.

$$\begin{aligned} r_{tt'} &= g(W^{(r)}z_{tt'} + b^{(r)}), \\ h_{tt'} &= o_t \odot \tanh(c_t \odot r_{tt'}) \end{aligned} \quad (3)$$

where  $g$  represents a non-linear activation function.

The standard LSTM gates  $i_t$ ,  $f_t$ ,  $o_t$ , and other vectors are computed using equation 1 from Section 2.1. The relation gating vector  $r_{tt'}$  in SRG-LSTM governs the degree of information from the cell state  $c_t$  that should be relayed to the hidden state  $h_t$  of unit  $t$ , and subsequently to the next unit  $t'$ . The control input  $z_{tt'}$  encapsulates the relationship between inputs  $x_t$  and  $x_{t'}$ . Differing from standard LSTMs, the hidden state in SRG-LSTM is influenced by the cell state, the output gate, and critically, the relation gate.

The relation gate is particularly advantageous in NLP contexts, where varying types of word relationships can significantly impact semantic composition. In our proposed SDT-LSTM model, the control input  $z_t$  indicates the dependency type between the word at node  $t$  and its corresponding headword at node  $t'$ . In a dependency parse tree, each node  $t$  typically has a single parent  $t'$ . However, in Dependency Acyclic Graphs (DAGs), a word can exhibit multiple dependency types with different words in the sentence. R-LSTM adapts to these scenarios by accommodating multiple control inputs and therefore, multiple hidden states, each tailored to the specific type of relation it embodies. For clarity, the relation gates are depicted separately, though in reality, a single relation gate and one relation weight matrix  $W^{(r)}$  exists in each SRG-LSTM unit. The relation gating vector  $r_{tt'}$  for each  $t' \in P(t)$ , the parents of node  $t$ , is calculated using equation 3.

#### 3.2. Syntactic Dependency Tree-LSTMs (SDT-LSTMs)

As previously discussed in section 2.2, Dependency Tree-LSTM, now termed as SDT-LSTM, uses identical weight sets across all LSTM units, disregarding the specific relation types between them. We introduce the SDT-LSTM model, which leverages R-LSTMs to learn embeddings representing various grammatical relations between word pairs. Our hypothesis is that this relational knowledge facilitates the construction of more nuanced semantic sentence representations. Similar to the original DT-LSTM by [7], the SDT-LSTM architecture adheres to the dependency tree topology, augmented with typed dependencies from the dependency parse.

Let  $D$  be an ordered universal set of typed dependencies in a language.

$$D = [d_1, d_2, \dots, d_l], l = |D| \quad (4)$$

Consider a sentence  $S = (w_1, w_2, \dots, w_n)$  comprising  $n$  words. The dependency parse of  $S$  is defined by a set of typed dependencies,

$$TD(S) = \{d_j(w_{t'}, w_t), d_j \in D, w_t \in S, w_{t'} \in S \cup \{ROOT\}\} \quad (5)$$

$TD(S)$  corresponds to a dependency tree with nodes representing words (excluding the root node) and edges labeled  $d_j$  linking node  $w_t$  to  $w_{t'}$  for each  $d_j(w_{t'}, w_t)$  in  $TD(S)$ . The tree's root is a special node  $ROOT$ , connected to the sentence's root word via an edge labeled 'root'.

Each SRG-LSTM<sup>(t)</sup> unit in the SDT-LSTM model aligns with a node  $t$  of the tree. The SRG-LSTM<sup>(t)</sup> inputs include: (1)  $x_t$ , the word vector for  $w_t$ , (2) the relation vector  $z_t = e^{(j)}$ , a binary vector with 1 at dimension  $j$  and 0 elsewhere if  $d_j(w_{t'}, w_t) \in TD(S)$ , and (3) outputs from its child SRG-LSTM units. The output of SRG-LSTM<sup>(t)</sup>, the cell state  $c_t$  and hidden state  $h_t$ , propagates to its parent SRG-LSTM<sup>(t')</sup>. In a dependency tree, any node  $t$  has only one parent, allowing us to omit the subscript  $t'$ . We combine equations 2 and 3 to formally define the SDT-LSTM as shown in equation 6

$$\begin{aligned} h_{c(t)} &= \sum_{k \in C(t)} h_k, \\ i_t &= \sigma(W^{(i)}x_t + U^{(i)}h_{c(t)} + b^{(i)}), \\ f_{tk} &= \sigma(W^{(f)}x_t + U^{(f)}h_k + b^{(f)}), \\ o_t &= \sigma(W^{(o)}x_t + U^{(o)}h_{c(t)} + b^{(o)}), \\ u_t &= \tanh(W^{(u)}x_t + U^{(u)}h_{c(t)} + b^{(u)}), \\ c_t &= i_t \odot u_t + \sum_{k \in C(t)} f_{tk} \odot c_k, \\ r_t &= g(W^{(r)}z_t + b^{(r)}), \\ h_t &= o_t \odot \tanh(c_t \odot r_t) \end{aligned} \quad (6)$$

The relation gating vector  $r_t$  for node  $t$  is derived using the one-hot encoding of the typed dependency  $z_t$  between node  $t$  and its parent in the dependency tree.

The weight matrix  $W^{(r)}$ , interpretable as task-specific embeddings for typed dependencies, allows each column  $w_j$  to correspond to a dependency type  $d_j$  from set  $D$ . These typed dependency embeddings are further explored in following section.

#### 4. Experiments

In this section, we assess the performance of our newly developed model, Semantic Relation-Gated Dependency Tree LSTM (SRG-DT-LSTM), across two primary tasks: (1) Semantic relatedness scoring of sentence pairs, and (2) Sentiment classification of movie reviews.

Our evaluation methodology aligns with the protocols established by [7]. The Stanford dependency parser [44] is employed to parse sentences in our training dataset. Word representations are initialized with pre-trained Glove embeddings [10]<sup>1</sup>. The model parameters are initialized randomly. We represent the 47 universal typed dependencies as one-hot vectors. The SRG-DT-LSTM model generates a distinctive embedding for each sentence. A softmax classifier is then used to derive semantic scores or sentiment labels from these sentence embeddings. The model is trained end-to-end, learning both sentence representation and semantic scores/sentiment labels. For semantic relatedness scoring, the word embeddings are non-trainable parameters.

<sup>1</sup> Glove: <http://nlp.stanford.edu/data/glove.42B.300d.zip>

#### 4.1. Semantic Relatedness Scoring

The goal of semantic relatedness scoring is to predict a numerical score reflecting the semantic similarity between two sentences. For each sentence pair  $(S_l, S_r)$  in the training set, the SRG-DT-LSTM model produces a corresponding vector pair  $(h_l, h_r)$ . We employ a Siamese architecture with shared model weights for both sentences in the pair. The element-wise product  $(h_l \odot h_r)$  and absolute difference  $(|h_l - h_r|)$  are fed into a neural network classifier that predicts the semantic similarity as a probability distribution  $\hat{p}$  across  $K$  classes using the softmax function. The predicted semantic score is calculated as follows:

$$\begin{aligned} h_s &= \sigma(U(h_l \odot h_r) + V(|h_l - h_r|) + b_h), \\ \hat{p}_\theta &= \text{softmax}(Wh_s + b_p), \\ \hat{y} &= r^T \hat{p}_\theta, \quad r = [1, 2, 3, \dots, K] \end{aligned} \quad (7)$$

To prepare the target probability distribution  $p$  from the actual similarity score  $y$ , where  $y = r^T p$ , each  $i^{\text{th}}$  element  $p_i$  in  $p$  is assigned as:

$$p_i = \begin{cases} 1 - |i - y|, & i \in [\lfloor y \rfloor, \lceil y \rceil] \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

The model is trained via back-propagation to minimize the regularized KL-divergence between  $p^{(j)}$  and  $\hat{p}_\theta^{(j)}$  for each sentence pair in the training set of size  $m$ .

$$J(\theta) = \frac{1}{m} \sum_{j=1}^m \text{KL}(p^{(j)} \parallel \hat{p}_\theta^{(j)}) + \frac{\lambda}{2} \|\theta\|_2^2, \quad (9)$$

#### 4.2. Sentiment Classification

Sentiment classification is modeled as a binary task, categorizing sentences as either positive or negative. The training dataset comprises dependency parse trees with certain nodes annotated with sentiment labels. Each node's label reflects the sentiment of the phrase it spans. For instance, in the review "It's not a great monster movie", the node covering "a great monster movie" is marked positive, while "not a great monster movie" is labeled negative. The overall review is deemed negative at the root node. The model predicts the root node's sentiment label recursively from its children.

The SRG-DT-LSTM model generates a hidden representation for every node in the dependency parse trees. This representation,  $h_t$ , abstractly signifies the sentiment of the phrase spanned by node  $t$ . At each node, a softmax classifier converts  $h_t$  into a probability distribution over  $K$  sentiment classes. The predicted class label for each node is the one with the highest probability.

$$\begin{aligned} p_\theta^{(t)}(y|\{x\}_t) &= \text{softmax}(Wh_t + b), \\ \hat{y}^{(t)} &= \text{argmax}_y p_\theta^{(t)}(y|\{x\}_t). \end{aligned} \quad (10)$$

The training aims to minimize the negative log-likelihood of the true class labels at each labeled node:

$$J(\theta) = -\frac{1}{m} \sum_{j=1}^m \log \hat{p}_\theta^{(j)}(y^{(j)}|\{x\}^{(j)}) + \frac{\lambda}{2} \|\theta\|_2^2, \quad (11)$$

### 4.3. Datasets and Training Details

For semantic relatedness scoring, we utilize the SICK dataset<sup>2</sup>, comprising 9927 sentence pairs, each annotated with a score from 1 to 5. The SICK dataset's sentences originate from image and video descriptions, scored by human evaluators. We adhere to the predefined train/dev/test split of 4500/500/4927.

For sentiment classification, we use the Stanford Sentiment Treebank (SST) [19]<sup>3</sup>. This dataset contains 11,855 sentences, each parsed to yield 215,154 unique phrases with manually assigned sentiment labels. We generated dependency trees for these sentences, labeling matching phrases accordingly. The train/dev/test split used for binary classification is 6920/872/1821 after excluding neutral sentences.

Table 1 lists the range of hyperparameters used, with optimal values highlighted. Training was conducted using an early stopping strategy with a patience setting of 10 epochs.

**Table 1.** Hyperparameter range used for model tuning, with the optimal values in **bold**.

Parameter	SICK-R	SST
Learning rate	0.01/0.05/0.1/0.2/ <b>0.25</b> /0.3	0.01/ <b>0.05</b> /0.1/0.2/0.25/0.3
Batch size	<b>25</b> /50/100	<b>25</b> /50/100
Memory dimension	120/ <b>150</b> /100	165/168/ <b>170</b> /175
Weight decay	0.0001	0.0001
Optimizer	<b>adagrad</b> /adam/nadam	<b>adagrad</b> /adam/nadam

## 5. Results and Discussion

### 5.1. Semantic Relatedness Assessment with SRG-LSTM and SDT-LSTM

In evaluating the efficacy of our SRG-LSTM and SDT-LSTM models for semantic relatedness scoring, we utilized the Pearson Correlation Coefficient and Mean-Squared Error (MSE) metrics to compare the actual relatedness score  $y$  with the predicted score  $\hat{y}$ . Our models were benchmarked against existing deep learning frameworks that employ LSTMs, dependency trees, or both, for semantic analysis. These frameworks are categorized as follows: the baseline mean vector approach, sequential models using LSTMs/GRUs (Gated Recurrent Units), tree-structured neural networks (tree-NNs) that leverage dependency trees, and models that integrate grammatical information into sequential models. SRG-LSTM and SDT-LSTM are part of the fourth category, utilizing both dependency structure and type.

Table 2 demonstrates that LSTM models surpass traditional RNN models, owing to the LSTM's superior sequence information retention capabilities. Hierarchical models like DT-LSTMs and DT-GRUs outperform sequential ones, underscoring the advantage of tree-structured representations in semantic understanding.

<sup>2</sup> SICK : <http://alt.qcri.org/semeval2014/task1/index.php?id=data-and-tools>

<sup>3</sup> SST dataset <http://nlp.stanford.edu/~socherr/stanfordSentimentTreebank.zip>

**Table 2.** Semantic relatedness scoring comparison of SRG-LSTM and SDT-LSTM with other LSTM models. Values are derived from existing literature.

Model	Pearson's $r$	MSE
Mean vectors	0.7577	0.4557
Sequential LSTM	0.8528	0.2831
Bidirectional LSTM (Bi-LSTM)	0.8567	0.2736
GRU Model	0.8595	0.2689
MaLSTM	0.8177	–
Dependency Tree-RNN (DT-RNN)	0.7923	0.3848
DT-LSTM	0.8676	0.2532
DT-GRU	0.8672	0.2573
Dependency-aware LSTM (D-LSTM)	0.8270	0.3527
Positional LSTM (pos-LSTM-n)	0.8263	–
Positional LSTM (pos-LSTM-v)	0.8149	–
Positional LSTM (pos-LSTM-nv)	0.8221	–
Positional LSTM (pos-LSTM-all)	0.8173	–
Semantic Dependency Tree-RNN (SDT-RNN)	0.7900	0.3848
<b>SRG-LSTM and SDT-LSTM</b>	<b>0.8731</b>	<b>0.2427</b>

The D-LSTM and various pos-LSTM models, which incorporate grammatical roles into sentence representation, showed enhancements over the MaLSTM model. Notably, the pos-LSTM-n, focusing solely on nouns, outperformed models considering a broader range of grammatical roles. This suggests that certain grammatical elements play a more pivotal role in semantic composition.

In our category, the SRG-LSTM and SDT-LSTM models, which consider both the structure and type of dependencies, show a marked improvement over other models, including the SDT-RNN. The SDT-RNN, originally designed for image captioning tasks, does not significantly outperform the DT-RNN in relatedness scoring, likely due to its complexity and the size of the network. Conversely, the SRG-LSTM and SDT-LSTM models, by representing each dependency type with a lower-dimensional vector, achieve better performance with fewer parameters, highlighting the efficiency of our approach.

Table 3 showcases how SRG-LSTM and SDT-LSTM retrieve the most semantically similar sentences for given queries from the SICK test dataset. The SRG-LSTM and SDT-LSTM models demonstrate a heightened ability to discern semantic nuances, as evidenced by their differentiated scoring for sentences with subtle variations.

**Table 3.** Illustrative examples of sentences retrieved by SRG-LSTM and SDT-LSTM from the SICK test set.  $S_{DT}$  denotes scores by Dependency Tree-LSTM, and  $S_{SRG-SDT}$  by SRG-LSTM and SDT-LSTM.

Query and Retrieved Sentences	$S_{DT}$	$S_{SRG-SDT}$
<b>Sample Query 1</b>		
Sentence A	4.48	4.81
Sentence B	4.48	4.66
Sentence C	4.48	4.57
<b>Sample Query 2</b>		
Sentence D	4.12	4.51
Sentence E	4.11	4.15
Sentence F	4.16	4.11
<b>Sample Query 3</b>		
Sentence G	4.79	4.87
Sentence H	4.82	4.88
Sentence I	4.85	4.87

Table 4 presents a selection of sentence pairs from the SICK dataset along with their corresponding actual (G) and predicted (S) similarity scores. This table illustrates the model's sensitivity to syntactic variations, showcasing its ability to maintain robustness against changes in sentence structure while accurately reflecting semantic shifts.

The detailed examination of these results, particularly in the context of SRG-LSTM and SDT-LSTM, underscores the significance of considering both dependency structure and type in semantic relatedness tasks. Our models' ability to efficiently integrate this information sets a new benchmark in the field, as reflected in the improved performance metrics.

**Table 4.** Selection of sentence pairs from the SICK test dataset with notable discrepancies between the predicted score (S) and ground truth (G).

Sentence 1	Sentence 2	S	G
A skateboarder is performing a stunt	A skateboarder is sliding on a rail	2.43	4.0
A basketball player is lying on the court with the ball being taken by another player	Two players are fighting for the ball on the basketball court	2.7	4.7
A motorcyclist is showing off tricks	The motorcyclist is being tricked by a performer	3.94	2.6
A gathering of five elderly people indoors	Five young people are hanging out indoors	4.64	3.4
A dog is leaping onto a diving board	A dog is jumping on a trampoline	4.12	2.9

Table 4 analyzes sentence pairs from the SICK dataset where the SRG-LSTM and SDT-LSTM's predicted scores show the greatest divergence from human ratings. In the first two pairs, the models underestimate the semantic relatedness, indicating a challenge in encoding subtle semantic nuances, such as understanding that *sliding on a rail* is a type of *skateboarding stunt*, or that *taking the ball* equates to *fighting for the ball* in basketball. The remaining pairs show overestimations of relatedness, suggesting that the models may overemphasize certain semantic similarities.

## 5.2. Typed Dependency Embeddings Analysis

As detailed in section 5.1, the SRG-LSTM and SDT-LSTM models incorporate specialized relation gates that learn task-specific embeddings for typed dependencies.

An analysis of the magnitudes of these typed dependency embeddings (Table 5) reveals their relative contributions to sentence meaning composition. This analysis confirms intuitive expectations: dependencies such as direct object, nominal modifier, adjectival modifier, and nominal subject are crucial, while others like goes-with and adjectival clause are less influential. Interestingly, the 'root' dependency does not rank at the top.

**Table 5.** Universal Dependencies ranked by their magnitude ( $M$ ) as learned by the SRG-LSTM and SDT-LSTM models.

Dependencies	Examples	Notation	$M$
Direct object	Chef <u>prepared</u> the <u>meal</u>	dobj(prepared,meal)	9.16
Nominal modifier	Meal was <u>cooked</u> by the <u>chef</u>	nmod(cooked,chef)	7.62
Adjectival modifier	Sam likes <u>spicy</u> <u>food</u>	amod(food,spicy)	7.31
Nominal subject	<u>Chef</u> <u>prepared</u> the meal	nsubj(prepared,chef)	7.27
Conjunction	Bill is <u>tall</u> and <u>kind</u>	conj(tall,kind)	6.97
Negation modifier	Bill is <u>not</u> a <u>scientist</u>	neg(scientist,not)	6.90
Case marking	I saw a cat <u>under</u> the <u>table</u>	case(table,under)	6.76

Examining sentence structures, we noticed that in active sentences, dependencies such as 'nsubj' and 'dobj' are prominent, while in passive constructions, these shift to 'nmod' and 'nsubjpass'. The model's embedding vectors

reflect a consistent pattern in these transformations, suggesting that SRG-LSTM and SDT-LSTM effectively capture changes in grammatical relationships across different sentence voices.

### 5.3. Sentiment Analysis Proficiency

Our evaluation of sentiment analysis capabilities (Table 6) indicates that the SRG-LSTM and SDT-LSTM models surpass standard LSTM and DT-LSTM models in accuracy. The performance is competitive with bidirectional LSTMs but slightly lower than that of CT-LSTMs (Constituency Tree-LSTMs).

**Table 6.** Performance comparison of SRG-LSTM and SDT-LSTM with other LSTM models for binary sentiment classification on the SST dataset. The values are sourced from prior studies.

Model	Accuracy(%)
LSTM	84.9
Bi-LSTM	87.5
2-layer LSTM	86.3
2-layer Bi-LSTM	87.2
CT-LSTM	88.0
DT-LSTM	85.7
<b>SRG-LSTM &amp; SDT-LSTM</b>	<b>86.9</b>

The SRG-LSTM and SDT-LSTM models demonstrate a Precision of 88.4%, Recall of 84.8%, and F1-score of 86.6% in detecting positive sentiments. These metrics outshine the DT-LSTM, which shows a Precision of 87.62%, Recall of 82.28%, and F1-score of 85.39%. This comparative analysis clearly illustrates the superiority of our models in sentiment recognition. The optimal hyperparameters for these models were determined empirically.

Table 7 presents examples from the SST test dataset, highlighting the accurate and erroneous sentiment predictions by the SRG-LSTM and SDT-LSTM models. The table categorizes predictions into true positives, true negatives, false positives, and false negatives, offering insights into the models' performance nuances.

**Table 7.** Sentiment analysis examples from the Stanford Sentiment Treebank (SST) with predicted sentiment score (S) and actual label (G). 0 denotes negative and 1 positive sentiment.

Input Sentences	S	G
"I wish I could enjoy the weekend, but I was relieved when it ended."	0	0
"Despite high expectations, the movie barely managed to move me."	0	0
"Quirky yet endearing, the film captures the essence of its theme."	1	1
"A deep dive into the complexities of love and sacrifice."	1	1
"Starts as an intriguing exploration but ends up as an underwhelming gimmick."	1	0
"Tedious for anyone except the most devoted fans."	1	0
"The film resonates with raw emotion, leaving a lasting impression."	0	1
"Subtle performances that deserve recognition and acclaim."	0	1

## 6. Conclusion and Prospects for Future Research

This study has yielded several key advancements in the field of natural language processing. Firstly, we have introduced the Syntactic Relation Gated LSTM (SRG-LSTM) architecture, a novel approach that enables the network to learn distinct gating vectors for various types of relationships in input sequences. Secondly, the Syntactic Dependency Tree LSTM (SDT-LSTM) model has been proposed, leveraging both the dependency parse structure and the grammatical relations between

words to enhance sentence semantic modeling. Finally, we have conducted a qualitative analysis exploring the impact of typed dependencies in language comprehension.

Our experimental results demonstrate that the SDT-LSTM model surpasses the DT-LSTM in both performance and learning efficiency for tasks such as semantic relatedness scoring and sentiment analysis. The SDT-LSTM exhibits a heightened ability to discern nuanced relationships within sentence pairs, achieving a closer alignment with human semantic interpretation compared to other contemporary methods.

The exploration of dependency types in semantic composition within a deep learning framework is a relatively uncharted territory. The computational models proposed in this research are poised to catalyze further investigations in this direction. Future endeavors will involve applying the SDT-LSTM model to a variety of NLP tasks, including paraphrase detection, natural language inference, question answering, and image captioning, where dependency tree-based models have shown significant promise.

A comprehensive analysis of the typed dependency embeddings learned by the SDT-LSTM model has unveiled intriguing insights into linguistic comprehension. These embeddings, from a linguistic standpoint, merit further exploration to deepen our understanding of language processing mechanisms.

In conclusion, the SRG-LSTM architecture introduced in this work presents a versatile concept that can be adapted to a range of domains. Given the prominence of LSTMs in numerous sequence modeling applications, the SRG-LSTM offers a compelling alternative for tasks that require modeling not just nodes but also the diverse types of links connecting them. Our findings suggest that SRG-LSTMs are adept at learning relationships between LSTM units, and the potential of incorporating a relation gate in other gated architectures, such as Tree-GRUs, is an exciting avenue for future exploration.

## References

1. Xia, W.; Zhu, W.; Liao, B.; Chen, M.; Cai, L.; Huang, L. Novel architecture for long short-term memory used in question classification. *Neurocomputing* **2018**, *299*, 20–31. doi:10.1016/j.neucom.2018.03.020.
2. Agarwal, B.; Ramampiaro, H.; Langseth, H.; Ruocco, M. A deep network model for paraphrase detection in short text messages. *Information Processing and Management* **2018**, *54*, 922–937. doi:10.1016/j.ipm.2018.06.005.
3. Jang, M.; Seo, S.; Kang, P. Recurrent neural network-based semantic variational autoencoder for Sequence-to-sequence learning. *Information Sciences* **2019**, *490*, 59–73. doi:10.1016/j.ins.2019.03.066.
4. Liu, Y.; Zhang, X.; Huang, F.; Tang, X.; Li, Z. Visual question answering via Attention-based syntactic structure tree-LSTM. *Applied Soft Computing* **2019**, *82*, 105584. doi:10.1016/j.asoc.2019.105584.
5. Fan, C.; Chen, W.; Wu, Y. Knowledge base question answering via path matching. *Knowledge-Based Systems* **2022**, *256*, 109857. doi:10.1016/j.knosys.2022.109857.
6. Kim, Y. Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, 2014. doi:10.3115/v1/d14-1181.
7. Tai, K.S.; Socher, R.; Manning, C.D. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, 2015. doi:10.3115/v1/p15-1150.
8. Shen, G.; Deng, Z.H.; Huang, T.; Chen, X. Learning to compose over tree structures via POS tags for sentence representation. *Expert Systems with Applications* **2020**, *141*, 112917. doi:10.1016/j.eswa.2019.112917.
9. Tien, N.H.; Le, N.M.; Tomohiro, Y.; Tatsuya, I. Sentence modeling via multiple word embeddings and multi-level comparison for semantic textual similarity. *Information Processing and Management* **2019**, *56*, 102090. doi:10.1016/j.ipm.2019.102090.
10. Pennington, J.; Socher, R.; Manning, C. Glove: Global vectors for word representation. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

11. Mitchell, J.; Lapata, M. Language models based on semantic composition. Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing Volume 1 - EMNLP '09. Association for Computational Linguistics, 2009, EMNLP '09. doi:10.3115/1699510.1699567.
12. Mueller, J.; Thyagarajan, A. Siamese Recurrent Architectures for Learning Sentence Similarity. *Proceedings of the AAAI Conference on Artificial Intelligence* **2016**, *30*. doi:10.1609/aaai.v30i1.10350.
13. Wang, Z.; Hamza, W.; Florian, R. Bilateral Multi-Perspective Matching for Natural Language Sentences. Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. International Joint Conferences on Artificial Intelligence Organization, 2017, IJCAI-2017. doi:10.24963/ijcai.2017/579.
14. Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A Convolutional Neural Network for Modelling Sentences. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, 2014. doi:10.3115/v1/p14-1062.
15. Socher, R.; Karpathy, A.; Le, Q.V.; Manning, C.D.; Ng, A.Y. Grounded Compositional Semantics for Finding and Describing Images with Sentences. *Transactions of the Association for Computational Linguistics* **2014**, *2*, 207–218. doi:10.1162/tacl\_a\_00177.
16. Fei, H.; Zhang, M.; Ji, D. Cross-Lingual Semantic Role Labeling with High-Quality Translated Training Corpus. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 7014–7026.
17. Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. Next-gpt: Any-to-any multimodal llm, 2023.
18. Fei, H.; Ren, Y.; Ji, D. Retrofitting Structure-aware Transformer Language Model for End Tasks. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 2020, pp. 2151–2161.
19. Wang, B.; Huang, J.; Zheng, H.; Wu, H. Semi-Supervised Recursive Autoencoders for Social Review Spam Detection. 2016 12th International Conference on Computational Intelligence and Security (CIS). IEEE, 2016. doi:10.1109/cis.2016.0035.
20. Choi, J.; Yoo, K.M.; Lee, S.g. Learning to Compose Task-Specific Tree Structures. *Proceedings of the AAAI Conference on Artificial Intelligence* **2018**, *32*. doi:10.1609/aaai.v32i1.11975.
21. Chen, D.; Manning, C. A Fast and Accurate Dependency Parser using Neural Networks. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, 2014. doi:10.3115/v1/d14-1082.
22. Wu, S.; Fei, H.; Ren, Y.; Ji, D.; Li, J. Learn from Syntax: Improving Pair-wise Aspect and Opinion Terms Extraction with Rich Syntactic Knowledge. Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, 2021, pp. 3957–3963.
23. Zhuang, L.; Fei, H.; Hu, P. Knowledge-enhanced event relation extraction via event ontology prompt. *Inf. Fusion* **2023**, *100*, 101919.
24. Wu, S.; Fei, H.; Li, F.; Zhang, M.; Liu, Y.; Teng, C.; Ji, D. Mastering the Explicit Opinion-Role Interaction: Syntax-Aided Neural Transition System for Unified Opinion Role Labeling. Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence, 2022, pp. 11513–11521.
25. Wu, S.; Fei, H.; Ji, W.; Chua, T.S. Cross2StrA: Unpaired Cross-lingual Image Captioning with Cross-lingual Cross-modal Structure-pivoted Alignment. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 2593–2608.
26. Fei, H.; Zhang, Y.; Ren, Y.; Ji, D. Latent Emotion Memory for Multi-Label Emotion Classification. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 7692–7699.
27. Manning, C.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.; McClosky, D. The Stanford CoreNLP Natural Language Processing Toolkit. Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. Association for Computational Linguistics, 2014. doi:10.3115/v1/p14-5010.
28. Kim, T.; Choi, J.; Edmiston, D.; Bae, S.; Lee, S.g. Dynamic Compositionality in Recursive Neural Networks with Structure-Aware Tag Representations. *Proceedings of the AAAI Conference on Artificial Intelligence* **2019**, *33*, 6594–6601. doi:10.1609/aaai.v33i01.33016594.
29. Fei, H.; Wu, S.; Li, J.; Li, B.; Li, F.; Qin, L.; Zhang, M.; Zhang, M.; Chua, T.S. LasUIE: Unifying Information Extraction with Latent Adaptive Structure-aware Generative Language Model. Proceedings of the Advances in Neural Information Processing Systems, NeurIPS 2022, 2022, pp. 15460–15475.

30. Liu, P.; Qiu, X.; Huang, X. Dynamic Compositional Neural Networks over Tree Structure. Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. International Joint Conferences on Artificial Intelligence Organization, 2017, IJCAI-2017. doi:10.24963/ijcai.2017/566.
31. Wang, J.; Lu, Z.; Song, G.; Fan, Y.; Du, L.; Lin, W. Tag2Vec: Learning Tag Representations in Tag Networks. The World Wide Web Conference. ACM, 2019, WWW '19. doi:10.1145/3308558.3313622.
32. Fei, H.; Li, F.; Li, B.; Ji, D. Encoder-Decoder Based Unified Semantic Role Labeling with Label-Aware Syntax. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, pp. 12794–12802.
33. Fei, H.; Ren, Y.; Zhang, Y.; Ji, D.; Liang, X. Enriching contextualized language model from knowledge graph for biomedical information extraction. *Briefings in Bioinformatics* **2021**, *22*.
34. Wu, S.; Fei, H.; Cao, Y.; Bing, L.; Chua, T.S. Information Screening whilst Exploiting! Multimodal Relation Extraction with Feature Denoising and Multimodal Topic Modeling. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 14734–14751.
35. Fei, H.; Ren, Y.; Ji, D. Boundaries and edges rethinking: An end-to-end neural model for overlapping entity relation extraction. *Information Processing & Management* **2020**, *57*, 102311.
36. Li, J.; Fei, H.; Liu, J.; Wu, S.; Zhang, M.; Teng, C.; Ji, D.; Li, F. Unified Named Entity Recognition as Word-Word Relation Classification. Proceedings of the AAAI Conference on Artificial Intelligence, 2022, pp. 10965–10973.
37. Li, J.; Xu, K.; Li, F.; Fei, H.; Ren, Y.; Ji, D. MRN: A Locally and Globally Mention-Based Reasoning Network for Document-Level Relation Extraction. Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, 2021, pp. 1359–1370.
38. Wang, F.; Li, F.; Fei, H.; Li, J.; Wu, S.; Su, F.; Shi, W.; Ji, D.; Cai, B. Entity-centered Cross-document Relation Extraction. Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, 2022, pp. 9871–9881.
39. Cao, H.; Li, J.; Su, F.; Li, F.; Fei, H.; Wu, S.; Li, B.; Zhao, L.; Ji, D. OneEE: A One-Stage Framework for Fast Overlapping and Nested Event Extraction. Proceedings of the 29th International Conference on Computational Linguistics, 2022, pp. 1953–1964.
40. Shi, W.; Li, F.; Li, J.; Fei, H.; Ji, D. Effective Token Graph Modeling using a Novel Labeling Strategy for Structured Sentiment Analysis. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 4232–4241.
41. Fei, H.; Wu, S.; Ren, Y.; Li, F.; Ji, D. Better Combine Them Together! Integrating Syntactic Constituency and Dependency Representations for Semantic Role Labeling. Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, 2021, pp. 549–559.
42. Zhu, W.; Yao, T.; Zhang, W.; Wei, B. Part-of-Speech-Based Long Short-Term Memory Network for Learning Sentence Representations. *IEEE Access* **2019**, *7*, 51810–51816. doi:10.1109/access.2019.2911983.
43. Fei, H.; Wu, S.; Ren, Y.; Zhang, M. Matching Structure for Dual Learning. Proceedings of the International Conference on Machine Learning, ICML, 2022, pp. 6373–6391.
44. Chen, D.; Manning, C. A fast and accurate dependency parser using neural networks. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 740–750.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.