Extracting Drug-Drug and Protein-Protein Interactions from Text Using a Continuous Update of Tree-Transformers

Sudipta Singha Roy
Department of Computer Science
The University of Western Ontario
London, ON, Canada
ssinghar@uwo.ca

Robert E. Mercer
Department of Computer Science
The University of Western Ontario
London, ON, Canada
mercer@csd.uwo.ca

Abstract

Understanding biological mechanisms requires determining mutual protein-protein interactions (PPI). Obtaining drug-drug interactions (DDI) from scientific articles provides important information about drugs. Extracting such medical entity interactions from biomedical articles is challenging due to complex sentence structures. To address this issue, our proposed model utilizes tree-transformers to generate the sentence representation first, and then a sentence-to-word update step to fine-tune the word embeddings which are again used by the tree-transformers to generate enriched sentence representations. Using the tree-transformers helps the model preserve syntactical information and provide semantic information. The fine-tuning provided by the continuous update step adds improved semantics to the representation of each sentence. Our model outperforms other prominent models with a significant performance boost on the five standard PPI corpora and a performance boost on the one benchmark DDI corpus that are used in our experiments.

1 Introduction

With the rapid expansion of scientific literature, most biological knowledge is now stored as text and can be accessed through scientific publications. The MEDLINE database has experienced a steady annual growth of over 4% for the last two decades, currently boasting a collection of over 29 million records from diverse sources. This is an increase of 3 million records compared to 2020 and over 8 million records compared to 2014, as cited in Yadav et al. (2020). The vast amount of textual data in biomedical research articles presents an invaluable opportunity for automated biomedical information retrieval to leverage this wealth of information.

As biomedical data continues to expand exponentially and the inherent complexity of textual representations, automated methods for information retrieval plays a pivotal role in aiding biologists in locating pertinent information, managing databases, and providing decision support to medical practitioners. Numerous studies have been conducted to extract valuable information from these texts, encompassing various domains such as protein-protein interactions, chemical-disease relationships, clinical correlations, drug-drug interactions, and more.

The internal biological processes within a cell, such as cellular organization, signal transduction, and immune response, are predominantly governed by interactions between different proteins (Sledzieski et al., 2021). To comprehend the molecular mechanisms underlying these biological processes, knowledge of protein-protein interactions (PPI) is indispensable (Ahmed et al., 2019a). These interactions have significant relevance in the biomedical domain, including drug target examination (Gordon et al., 2020) and signal proteins (Altmann et al., 2020). Consequently, the identification of protein-protein interactions (PPIs) leads to a deeper understanding of the functions, regulation, and communication between various proteins (Yao et al., 2019). The primary objective of PPI recognition is to extract the relationships between protein entities mentioned in a document (Krallinger et al., 2008).

A drug-drug interaction (DDI) refers to a modification in the effects of one drug due to the presence of another drug (Rodrigues, 2019). While clinical trials for pre-market identification of interactions are challenging, obtaining DDI information from scientific articles is a faster, cost-effective, and reliable approach to reducing adverse effects. Furthermore, in order to practice evidence-based medicine and mitigate drug-related accidents, comprehensive extraction of DDI knowledge from pharmaceutical literature is crucial (Sackett, 1997). Automatic DDI extraction can prove highly beneficial for the pharmaceutical industry, offering an efficient means of reducing the time spent by healthcare professionals

The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks, pages 280–291
July 13, 2023 ©2023 Association for Computational Linguistics

reviewing the medical literature.

Biomedical literature contains a wealth of information about protein-protein interactions (PPIs) and drug-drug interactions (DDIs), but this information is often unstructured. Manual extraction of these interactions from biomedical literature is a laborious, resource-intensive, and costly task, given the sheer volume of published studies (Peng et al., 2016; Tang et al., 2022). Consequently, the automatic extraction of PPIs and DDIs from biomedical literature has emerged as a vital research area, garnering attention from numerous researchers. While the information may be scattered throughout the document, the current study focuses on detecting these interactions within individual sentences, similar to previous studies (Asada et al., 2023; Fei et al., 2021; Ahmed et al., 2019a; Tikk et al., 2010).

An instance of a sentence that demonstrates protein-protein interactions can be found in the study by Howard et al. (2000), where it states:

> "At 89.3 nmol/L, maximal migration of CCR1 and CCR8 transfected cells was prompted by LEC and at 5.6 nmol/L, cell adhesion also occurred."

This sentence highlights two protein-protein interactions involving LEC and CCR1, as well as LEC and CCR8. However, it is important to note that there is no correlation mentioned between proteins CCR1 and CCR8 in this context.

An instance of a DDI-containing sentence is (Nauta et al., 1974):

> "To determine whether probenecid has a direct effect on the distribution of cloxacillin, the elimination and distribution of cloxacillin was studied in six patients, five lacking kidney function and one with a partially impaired renal function, in the presence or absence of probenecid."

This sentence mentions two drugs: probenecid, and cloxacillin. However, the interaction between them is negative, as no concrete interaction is stated.

During the extraction of relationships between target proteins or drugs, we have addressed three key concerns. Firstly, how to tackle the challenge of retrieving relations when the mentioned proteins or drugs are widely separated in the text. Secondly, how to preserve better semantics by handling the phrasal structure of the text, allowing for the effective extraction of PPIs or DDIs and capturing relevant information. Lastly, what is the impact of updating word and non-leaf node representations in the tree-structured networks based on the sentence at hand, as opposed to using fixed representations from pre-trained models, and how this influences the generated sentence representation for the task of PPI and DDI extractions.

To address the above-mentioned three considerations we have proposed a model combining a constituency tree-transformer (for preserving phrase-level information in the text), and a dependency tree-transformer (to consider relations between long distant drugs or proteins in the text) where each of them generates sentence representations which are then combined. Finally, a sentence-to-word update step is introduced following the concept from Wang et al. (2020) to update the word and non-leaf nodes of the tree-transformers to generate refined sentence representation. This approach serves the purpose of fine-tuning BERT-based word embeddings for these tasks. But the advantage of this approach is that we do not need to fine-tune millions of parameters in the BERT-based models. Our study includes a thorough analysis of the performance of the proposed models on benchmark PPI and DDI datasets. The results demonstrate the superiority of our proposed model compared to previous prominent models in the field. The comprehensive analysis highlights the effectiveness and efficacy of our approach in accurately extracting protein-protein and drug-drug interactions from biomedical literature.[1]

2 Related Work

In the initial stages of biomedical entity relation extraction research, co-occurrence and pattern recognition techniques were commonly employed (Baumgartner et al., 2008; Yu et al., 2018). However, with advancements in technology, machine learning techniques have gained prominence due to their superior performance. Early approaches involved feature engineering and kernel methods to construct a feature set, followed by classification using support vector machines or other classifiers (Airola et al., 2008; Murugesan et al., 2017). In recent years, deep learning techniques, leveraging the widespread use of deep learning in natural language processing (NLP), have been successfully applied to PPI and DDI extraction in several re-

[1] The code is available on https://github.com/sudipta90/BioNLP_PPI_Heterogeneous.git

search works (Liu et al., 2016; Zhao et al., 2016; Hua and Quan, 2016; Choi, 2016). Zhang et al. (2018) proposed a three-channel convolution neural network for extracting PPIs from the text.

Recent work in PPI and DDI extraction often utilizes recurrent neural network (RNN) models that treat textual representations as sequences (Hsieh et al., 2017; Sahu and Anand, 2018; Yadav et al., 2019). However, these models may miss semantic compositions when biomedical entities lie at distant positions in the text, as they only consider word order and ignore linguistic structure (Ahmed et al., 2019b; Li et al., 2015). In contrast, recursive neural networks, also known as tree-structured neural network models (Ahmed et al., 2019c; Singha Roy and Mercer, 2022), process sentences represented in a parsed tree form, capturing both syntax and semantics in a more effective manner. There have also been investigations into graph-based methods, where models operate on a fully connected graph composed of either word or phrase nodes (Fei et al., 2021). These approaches aim to leverage the structural information present in the data for improved performance in PPI and DDI extraction. Asada et al. (2021) utilized molecular structure and description of the drugs for retrieving DDIs. Gu et al. (2021) fine-tuned PubMedBERT to extract relations between drugs. Following this, Asada et al. (2023) utilized a knowledge graph with PubMedBERT for the DDI extraction task.

3 Proposed Model

In this section, we provide details of our model for the protein-protein and drug-drug interaction extraction tasks. Our model contains three key modules: two tree-transformers, as described in Ahmed et al. (2019c), for preserving the semantic and syntactical information, and a sentence-to-word update step for updating the word and intermediate node representations in the tree-transformers to generate refined representations of the sentences. In this current work, we have added an update of the word embeddings after the sentence-to-word update step which enriches the input to the combination of the two tree-transformers and the heterogeneous graph attention network, which were first proposed for the PPI extraction task in Singha Roy and Mercer (2023). In this section, we first discuss how each module functions individually, and then elaborate on how these modules are integrated into our proposed model with the expanded workflow.

3.1 Tree-Transformers

The two tree-based representations commonly used for representing a sentence are constituency trees and dependency trees. Constituency trees capture the structure of phrases in a sentence, while dependency trees represent the dependencies between individual words. In our work, we utilize two tree-transformer models, namely the dependency tree-transformer and the constituency tree-transformer, as proposed by Ahmed et al. (2019c), to leverage these sources of syntactic structure information. The goal of these tree-transformer models is to traverse each sub-tree within a dependency or constituency tree structure attentively and at its root derive a sentence representation. This allows us to capture both the semantic and syntactic information of the sentence for improved performance in extracting protein-protein and drug-drug interactions from the text. Unlike the tree transformer proposed byWang et al. (2019) which learns phrases they call constituents, the tree transformer proposed by Ahmed et al. (2019c) works over the parsed trees and can work with both the constituency and dependency trees.

In a dependency tree, each node represents a word in the sentence. When traversing a sub-tree in a dependency tree, the dependency tree-transformer takes into consideration the representations of both the parent and child nodes, allowing for the propagation of information between connected words in the tree. On the other hand, in a constituency tree, only the leaf nodes hold words, while the non-terminal nodes do not have word representations. The vector representations for the non-terminal nodes are computed only after the sub-tree has been fully traversed, taking into account the information from the leaf nodes. This approach allows for the capture of both local and global contextual information during the tree traversal process, facilitating the extraction of meaningful representations from the syntactic structures of the sentence. Ahmed et al. (2019c) have used a self-attention mechanism to process the dependency and constituency tree representations of the sentence, employing query ($\mathcal{Q}$), key ($\mathcal{K}$), and value ($\mathcal{V}$) matrices, which are computed as follows based on the formulation proposed by Vaswani et al. (Vaswani et al., 2017):

$$\mathcal{K} = \omega_k \mathcal{M}_k \quad \text{s.t.} \quad \omega_k \in \mathbb{R}^{d \times d} \tag{1}$$

$$\mathcal{V} = \omega_v \mathcal{M}_v \quad \text{s.t.} \quad \omega_v \in \mathbb{R}^{d \times d} \tag{2}$$

$$\mathcal{Q} = \omega_q \mathcal{M}_q \quad \text{s.t.} \quad \omega_q \in \mathbb{R}^{d \times d} \tag{3}$$

In the tree-based transformer models, the matrix $\mathcal{M}$ is computed differently for dependency trees and constituency trees. In the case of dependency trees, the matrix $\mathcal{M}$ is created by concatenating the word vectors of all of the child nodes for each parent node in the dependency tree. On the other hand, for constituency trees, $\mathcal{M}$ is formed by concatenating the word vectors within a constituent. The self-attention matrix α is then computed as:

$$\alpha = \text{softmax}(\frac{\mathcal{Q}\,\mathcal{K}^T}{\sqrt{d_k}})\mathcal{V} \tag{4}$$

where d_k represents the dimension of $\mathcal{K}$. To implement multi-branch attention with n branches, the following steps are taken: first, n copies of the key, query, and value matrices are generated using weight matrices (ω_i). Then, each branch applies the scaled dot product attention separately (following Eq. 4), using its own set of query, key, and value vectors. Finally, this results in n sets of attended word vectors, one for each branch (see Eq. 5).

$$\mathcal{B}_i = \alpha_{i\in[1,n]}(\mathcal{Q}_i\,\omega_i^{\mathcal{Q}}, \mathcal{K}_i\,\omega_i^{\mathcal{K}}, \mathcal{V}_i\,\omega_i^{\mathcal{V}}) \tag{5}$$

Then, a residual connection is employed on these tensors, and a batch normalization layer is applied to each layer. Following that, the branch representation is generated using a scaling factor μ in the following manner:

$$\tilde{\mathcal{B}}_i = \texttt{LayerNorm}(\mathcal{B}_i\omega_i^b + \mathcal{B}_i) \times \mu_i \tag{6}$$

Following that, a position-wise CNN (PCNN) is applied to each $\tilde{\mathcal{B}}_i$. The PCNN layer comprises two convolution operations on each position, separated by a rectified linear unit (ReLU) activation function. The operation of this PCNN layer can be represented as per Eq. 7:

$$\texttt{PCNN}(x) = \texttt{Conv}(\texttt{ReLU}(\texttt{Conv}(x) + b_1)) + b_2 \tag{7}$$

The ultimate attentive representation of these semantic sub-spaces, which are generated from the PCNN layer, is acquired by conducting a linear weighted summation (as expressed in Eq. 8), with $\gamma \in \mathbb{R}^n$ serving as a hyper-parameter of the model.

$$\texttt{BranchAttn} = \sum_{i=1}^{n} \gamma_i \texttt{PCNN}(\tilde{\mathcal{B}}_i) \tag{8}$$

In the final stage, a residual connection is established with `BranchAttn`, and a hyperbolic tangent non-linearity (tanh) function is applied. The representation of the parent node is then obtained by conducting element-wise summation (`EWS`) (Eq. 9).

$$\texttt{ParentNode} = \texttt{EWS}(\tanh((\chi_{attn} + \chi)\omega + b)) \tag{9}$$

In Eq. 9, the symbols χ and χ_{attn} represent the input and output features of the attention computation module, respectively.

3.2 Sentence-to-Word Update Module

For the sentence-to-word update step, we have used an approach similar to the heterogeneous graph attention network (H-GAT) (Wang et al., 2020). H-GAT was introduced for extractive summarization tasks with the intention to generate an enriched cross-sentence relationship. In our research, we have employed this approach to enhance the quality of sentence representations. This module is utilized at each iteration, once the forward passes of the constituency and dependency tree-transformers are completed. Through sentence-to-word and a following forward pass of the tree-transformers again, this module enriches the sentence vectors, thereby improving the overall sentence representation quality.

The graph G in this module is structured as $G = V, E$, where V represents the nodes in the graph and E represents the edges between those nodes. For a given sentence S containing n words (w_i), the set of nodes V is defined as $V = w_1, w_2, ..., w_n, S$. Since the task involves identifying PPIs and DDIs in single sentences, the edges are established in such a way that the sentence node S is connected to every word node w_i. Once the graph G is constructed, a Graph Attention Network (GAT) (Veličković et al., 2018) is used to modify the feature values of the nodes. Let $h_i \in \mathbb{R}^{d_h}$ be the hidden states of the word and sentence nodes, where $i \in 1 : (n + 1)$ and d_h is the hidden state dimension. The GAT layer can be formulated as follows:

$$\kappa_{i,j} = \texttt{LeakyReLU}(\omega_a[\omega_q h_i; \omega_k h_j]) \tag{10}$$

$$\alpha_{i,j} = \frac{\exp(\kappa_{i,j})}{\sum_{l\in\mathcal{N}_i}\exp(\kappa_{i,l})} \tag{11}$$

$$\mathcal{Z}_i = \sigma(\sum_{j\in\mathcal{N}_i} \alpha_{i,j}\omega_v h_j) \tag{12}$$

The weight matrices ω_a, ω_q, ω_k, and ω_v in the GAT layer are updated through backpropagation. The set of neighbouring nodes for a given node i is denoted by $\mathcal{N}i$, while the attention score between hidden states h_i and h_j is denoted by $\alpha_{i,j}$. The GAT layer can be extended to incorporate multi-head attention with $\mathcal{M}$ heads, which is represented as follows:

$$\mathcal{Z}^i = ||_{m=1}^{\mathcal{M}} \sigma(\sum_{j \in \mathcal{N}_i} \alpha_{i,j}^m \omega^m h_i) \tag{13}$$

To mitigate the issue of vanishing gradients over time, a residual connection is established. The final hidden state representation (h_i), incorporating the information (u_i) from the residual connection, is formulated as $h_i = u_i + h_i$.

The word nodes are updated using the previously delineated GAT and a position-wise feed-forward network (FFN) layer, which consists of two linear transformations as introduced by Wang et al. (2020). At the t-th iteration, the updates are performed based on the information from the sentence node, as shown in Eqs. 14 and 15:

$$\mathcal{Z}_{s \to w}^{t+1} = GAT(\mathcal{H}_w^t, \mathcal{H}_s^t, \mathcal{H}_s^t) \tag{14}$$

$$\mathcal{H}_w^{t+1} = FFN(\mathcal{Z}_{s \to w}^{t+1} + \mathcal{H}_s^t) \tag{15}$$

In Eq. 14, $\mathcal{H}_w^0$ represents the set of word nodes, which are the Bio-RoBERTa-based embeddings for the words in the sentence (Gururangan et al., 2020). On the other hand, $\mathcal{H}_s^t$ represents the average of the sentence representations obtained from the dependency and constituency tree-transformers. In the GAT layer, $\mathcal{H}_w^t$ is used as the query, while $\mathcal{H}_s^t$ is considered as both the value and key matrices, imitating the approach of Vaswani et al. (2017).

3.3 Model Architecture

Figure 1 provides an architectural overview of the model. The model starts with Bio-RoBERTa word embeddings as input. These embeddings are then processed by the Dependency Tree Transformer (DTT) and Constituency Tree Transformer (CTT) in parallel to generate sentence representations (S_{DTT} and S_{CTT}, accordingly). This step is followed by a mean-pooling operation and an intermediate sentence representation S_{avg} is generated. The sentence-to-word update step uses the S_{avg} representation to update the word representations. These updated word representations are then passed to the tree-transformers again. This step involves another forward pass to generate the updated sentence representations S'_{DTT} and S'_{CTT}. Max-pooling is applied over these updated sentence representations and this result is fed to the following classification layer for the relation extraction.

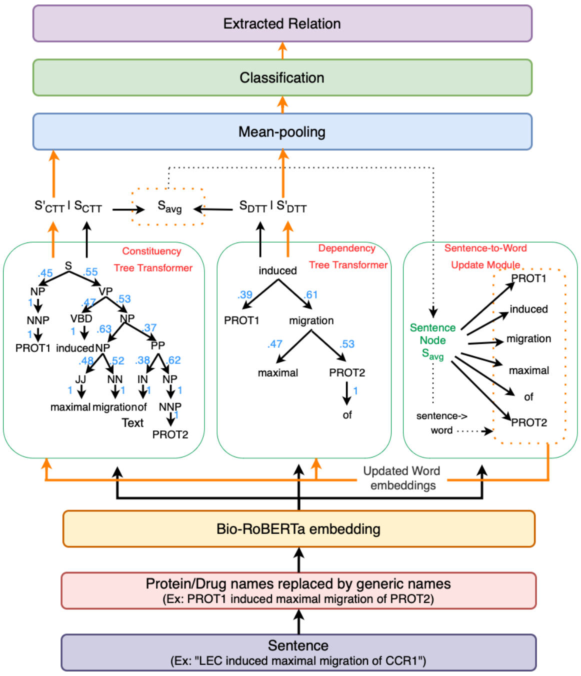

Figure 1: Integrated architecture with tree-transformers with the sentence-to-word update step for relation extraction task. The numerical values in blue color, associated with the branches in the tree Transformers, represent the attention scores for those specific branches.

4 Experimental Setup and Analysis of Results

In this section, the performance of the proposed model is evaluated using the F1-score. The PPI (Protein-Protein Interaction) and DDI (Drug-Drug Interaction) extraction tasks have been formulated as classification tasks. The section also includes a demographical overview of the five primary PPI corpora and the standard DDI corpus used in the evaluation, as well as a discussion of the preprocessing techniques employed on these corpora. The efficacy of the proposed model is compared to leading sequential, tree-structured, and graph-based architectures that have been previously propounded for these biomedical entity inter-relation extraction tasks.

4.1 Corpora Descriptions

The performance of the proposed model for PPI extraction task is evaluated on five benchmark corpora: BioInfer (Pyysalo et al., 2007), AIMed (Bunescu et al., 2005), HPRD50 (Fundel et al., 2007), IEPA (Ding et al., 2001), and LLL (Nédellec, 2005). In order to bring forth a persistent classification task across all five corpora, protein names are substituted with three symbols: PROTEIN1 and

PROTEIN2 are used to represent pairs of proteins that are considered potentially interacting in a given sentence, while all other protein names present in the sentence are altered with PROTEIN0. The approach of replacing protein names with generic symbols allows the model to focus on the interaction between a pair of proteins in each sentence, one at a time. For sentences containing more than two proteins, two proteins at a time are tagged with PROTEIN1 and PROTEIN2, and their interaction (positive or negative) is identified. This process is repeated sequentially for all protein pairs in the sentence. Thus, for each sentence in the corpus containing η proteins, the modified corpus will feature $^{\eta}C_2$ variations. For example, consider the sentence: "At 89.3 nmol/L, maximal migration of CCR1 and CCR8 transfected cells was prompted by LEC and at 5.6 nmol/L, cell adhesion also occurred." To identify the possible relationship between LEC and CCR1, their respective protein names are replaced with PROTEIN1 and PROTEIN2, while CCR8 is replaced with PROTEIN0. When the objective is to identify the possible interaction between LEC and CCR8, their names are replaced with PROTEIN1 and PROTEIN2, and PROTEIN0 is used in place of CCR1. Similarly, when identifying the possible interaction between CCR1 and CCR8, they are replaced with PROTEIN1 and PROTEIN2, and LEC is replaced with PROTEIN0. Interactions between protein pairs can be either positive or negative. For the above example, when the considered proteins are CCR1 and LEC or CCR8 and LEC, the nature of their interactions is positive in each case. However, when the considered protein pair is CCR1 and CCR8, the PPI is negative since no interaction is present between them. Thus, the example sentence presents three possible interactions, resulting in three variants ($^{3}C_2$) of the sentence in the modified corpus: two with positive interactions and one with a negative interaction. Using generic names to represent protein names enhances the data by allowing for multiple samples of these generic names, as opposed to only a few samples for each individual protein name. An overview of the demographic traits for the five revised datasets, using the aforementioned method, is presented in Table 1.

For the DDI extraction task, we have conducted our experiments on the DDIExtraction-2013 corpus (Segura-Bedmar et al., 2013). For the data preprocessing step, the aforementioned steps have been similarly followed. Here, the potentially interacting drug pairs are replaced with DRUG1 and DRUG2 and the remaining drug names in the sentence are replaced with DRUG0. Thus, each sample considers the interaction between one pair of drugs at a time, similar to the PPI data preprocessing step. The overall demographic of the corpus is presented in Table 2.

Table 1: Demographical description of the modified corpora for PPI task

Corpus	Original Sentences	Positive Samples	Negative Samples
AIMED	1,995	1,000	4,834
BioInfer	1,100	2,534	7,132
IEPA	486	335	482
HPRD50	145	163	270
LLL	77	164	166

Table 2: Demographical description of the SemEval-2013 DDIExtraction task dataset

	Train	Test
Sentences	6976	1299
Drug Pairs	27792	5716
Positive Pairs	4021	979
Mechanism	1319	302
Effect	1687	360
Advice	826	221
Interaction	189	96
Negative Pairs	23771	4737

The Stanford dependency and constituency parsers (Manning et al., 2014) have been employed to parse sentences in all of these corpora.

4.2 Experimental Setup

Regarding the model specifics, an initial learning rate of 0.1 has been employed for all the experiments. If the validation accuracy declines compared to the previous iteration, the learning rate has been decreased by 80% in each subsequent iteration. Additionally, a batch size of 10 is set.

The tree-transformer models incorporate six branches of an attention layer and six PCNN layers. Two CNN layers utilize kernels of dimensions 341 and 300, respectively, with a dropout of 0.1 in the second layer only. The sentence-to-word

Table 3: Performance evaluation of the models for PPI extraction on the five datasets: F1-score (in %) as the metric. All values, except for Tai et al. (2015) and the Proposed Model, are those reported in the original works. The best performance metric for each dataset is indicated in **bold**.

Methods	Architecture	AIMed	BioInfer	IEPA	HPRD50	LLL	Avg.
Chang et al. (2016)	RNN	60.6	69.4	71.4	71.5	80.6	70.7
Hsieh et al. (2017)	RNN	76.9	87.2	76.31	80.51	78.3	79.84
Zhang et al. (2018)	RNN	56.4	61.3	75.1	63.4	76.5	66.54
Yadav et al. (2020)	RNN	77.33	76.33	-	-	-	76.83
Tai et al. (2015)	Tree-structured	80.6	88.1	76.4	82.0	84.8	82.38
Ahmed et al. (2019a)	Tree-structured	81.6	89.1	78.5	81.3	84.2	82.94
Singha Roy and Mercer (2022)	Tree-structured	88.15	96.01	83.24	88.94	92.18	89.70
Fei et al. (2021)	Graph-based	88.27	96.21	83.90	89.57	92.86	90.16
Singha Roy and Mercer (2023)	Tree-structured + Heterogeneous Graph	91.23	96.97	87.28	93.11	93.52	92.02
Proposed Model	Tree-structured + Heterogeneous Graph	**94.66**	**97.81**	**93.47**	**94.01**	**94.14**	**94.82**

update module employs six attention heads. The trainable hyperparameters of the model are updated using the Adagrad optimizer (Lydia and Francis, 2019). The final representation for each sentence representation unit (dependency and constituency tree-transformers) and the model itself is a 512-dimensional vector. Bio-RoBERTa word embeddings are used as the initial input of the model. The model uses two forward passes for sentence vector generation. Only the first forward pass uses these Bio-RoBERTa word embeddings. The second pass utilizes the updated word representations obtained from the sentence-to-word update module, as described in Section 3.3.

To conduct the performance evaluation of the Proposed Model for the PPI extraction task, we have employed StratifiedK-Fold from the scikit-learn package to perform 10-fold cross-validation. In each fold, the training has been carried out on the training set, and the evaluation has been performed on a separate test set. The tree LSTM proposed by Tai et al.[2] (2015) has been trained and tested by us following the aforementioned approach. All the other models' results are reported directly from their corresponding publications. For the DDI extraction task, the training and test sets have been shuffled 5 times using StratifiedK-Fold from the scikit-learn package to perform 5-fold cross-validation. The average performance metrics for both tasks are presented in Tables 3 and 4, respectively, and discussed in Section 4.3.

The experiments have been conducted on a Linux Ubuntu 22.04 LTE machine equipped with 16GB of memory and an Nvidia 1070Ti graphics card with 8GB of graphics memory. PyTorch 1.7.1 has been utilized for implementing the model.

4.3 Performance Analysis

Table 3 showcases the performance of our proposed model on the five benchmark corpora for PPI extraction, along with the published results of various sequential, tree-structured, and graph-based models for comparison. The F1-score has been utilized as the performance evaluation metric.

Our proposed model has demonstrated outstanding performance on all benchmark corpora, particularly on the AIMED, IEPA and BioInfer datasets. For the AIMED corpus, our model has achieved an impressive F1-score of 94.66%, surpassing the current state-of-the-art (SOTA) model (Fei et al., 2021) by 6.39 percentage points (p.p.). For the BioInfer dataset, which has longer sentences and more protein names mentioned in a single sentence, our model has shown remarkable results achieving an F1-score of 97.81%, surpassing the SOTA and Singha Roy and Mercer (2022) results by 1.6 p.p. and 1.8 p.p., respectively. Even for the IEPA, HPRD50, and LLL corpora, which have smaller sample sizes, our model has outperformed the current SOTA. Compared to the best performing tree-structured model (Singha Roy and Mercer, 2022), our model has achieved significant improvements of 10.23 p.p., 5.07 p.p., and 1.96 p.p. for the IEPA, HPRD50, and LLL corpora, respectively. In comparison to Fei et al. (2021), our model has achieved performance boosts of 9.57 p.p., 4.44 p.p., and 1.96 p.p. for the same three corpora, respectively. On average, across all five corpora, our model has

[2] This model was not developed in particular for the PPI task. We were interested in its performance on this task.

Table 4: Performance evaluation of the models on SemEval-2013 DDIExtraction: **P**recision, **R**ecall, and **F1**-score (in %) as the metrics. All values, except for the Proposed Model, are those reported in the original works. The best performance metrics are indicated in **bold**.

Methods	Architecture	P	R	F1
Yadav et al. (2019)	RNN	76.5	69.0	72.6
Gu et al. (2021)	PubMedBERT	-	-	82.4
Phan et al. (2021)	RNN	-	-	83.7
Asada et al. (2021)	Knowledge-based	85.4	82.8	84.1
Asada et al. (2023)	PubMedBERT + Knowledge	85.3	85.5	85.4
Fei et al. (2021)	Graph-based	94.9	92.0	93.4
Proposed Model	Tree-structured + Heterogeneous Graph	**95.5**	**94.9**	**95.2**

Table 5: Performance of the model on individual DDI types of the SemEval-2013 DDIExtraction dataset

Metric	Mech.	Effect	Advice	Interac.
P	95.83	96.77	95.10	94.33
R	94.27	95.64	94.89	94.61
F1	95.04	96.20	94.99	94.47

obtained an impressive F1-score of 94.82%, surpassing the results reported in Fei et al. (2021) by 4.66 p.p.

Table 4 shows the precision (P), recall (R) and F1-score achieved by the proposed model for the DDI extraction task over the SemEval-2013 DDIExtraction corpus along with previous prominent models and Table 5 portrays the performance of the model over each individual class of the corpus. From Table 4 it is clearly visible that the proposed model has outperformed the current SOTA (Fei et al., 2021) with a significant margin of 1.8 p.p. by achieving 95.2% F1-score. For each individual type, the model has achieved more than 94% F1-score which also proves the generalization capability of the proposed model.

The first attempt to extract PPIs from text incorporating a tree structured neural network model was by Ahmed et al. (2019a). They have applied structured attention over tree-LSTMs and achieved an average of 82.94% F1-score over the 5 benchmark PPI corpora. Later, in our following work (Singha Roy and Mercer, 2022), we have applied tree-transformers and gained a 6.86 p.p. performance boost on average. This model almost reached Fei et al.'s (2021) work which was the state-of-the-art at that time. In the next step, we have experimented with adding an heterogeneous graph attention network model (Singha Roy and Mercer, 2023) with the tree transformers and observed a further performance gain of 2.32 p.p. In the work reported here, we have utilized the same heterogeneous graph attention network to update the word embeddings to generate a refined sentence vector which has given us another 2.8 p.p. performance gain over the PPI corpora, giving a total improvement of 5.12 p.p. from our initial tree-transformer model (Singha Roy and Mercer, 2022). In this present work we have also experimented with the DDI corpus to show the generalizability of the method and gained a 1.8 p.p. F1-score improvement over the previous state-of-the-art (Fei et al., 2021).

4.4 Ablation Study

To indicate the importance of each module in the Proposed Model, an ablation study has been performed and the results are presented in Table 6.

If the sentence-to-word update module is discarded the model is similar to the work of Singha Roy and Mercer (2022) and we can see a significant drop in the F1-score when this module is discarded. For the five PPI extraction corpora, this F1-score drop is 5.12 p.p. on average. For the SemEval-2013 DDIExtraction dataset (mentioned as DDI in Table 6), this F1-score drop is 5.1 p.p. which reflects the effectiveness of the sentence-to-word update module. This process plays a critical role in capturing relevant contextual information from both the sentence and word levels, leading to the enhanced model performance.

We believe that the improved performance is due to the sentence-to-word update module leveraging the sentence representations generated by the tree-transformers fed with task-specific and context-enriched word vectors. These sentence

Table 6: The ablation study of the Proposed Model on the PPI and DDI corpora. All values are F1-scores.

Discarded Component	AIMed	BioInfer	IEPA	HPRD50	LLL	DDI
Constituency Tree-Transformer	89.32	95.66	85.82	90.46	92.01	91.63
Dependency Tree-Transformer	89.11	95.43	84.60	89.72	91.78	90.96
Sentence-to-Word Update Module	88.11	95.89	83.17	88.85	92.10	89.98

representations, along with the newly generated word representations through the sentence-to-word update step, enrich the semantics of the task. Consequently, the second forward pass produces a more informative sentence representation for the subsequent classifier, contributing to the enhanced performance of our model.

The significance of the sentence-to-word update module is also supported by the other two ablation experiments presented in the table. When only one of the tree-transformers is utilized with the sentence-to-word update module, it performs better than that individual tree-transformer for these tasks. As reported in Singha Roy and Mercer (2022), the dependency tree-transformer achieves 89.06% F1-score over the PPI extraction corpora on average, where with the sentence-to-word update module it is 90.65%. In the case of the constituency tree-transformer, the performance boost is 1.33 p.p. A similar observation has been found for the experiments with the DDI corpus, as well.

5 Conclusions and Future Work

From these results and discussions in the previous sections, we can conclude that our model performs significantly better than the other prominent models even without using any additional features. The tree-transformers enable the proposed model to capture better semantics along with syntactical information. Additionally, the sentence-to-word update module provides more task-specific context-aware information, generating enriched word embeddings that further enhance the sentence representations for the PPI and DDI extraction tasks.

Although the model has achieved a significantly improved performance over the previous models, still there is scope for further improvement. Including a knowledge-graph, like in Asada et al. (2023), may improve the model performance with proper knowledge about the DDI extraction task. Moreover, the current models find PPIs and DDIs that are given in a single sentence. Using an additional layer of hierarchy that represents document-to-sentence relations over the sentence-to-word update module, this work can be extended to extract relations between biomedical entities lying in different sentences.

Limitation

The model has achieved a significant performance boost. However, the trade-off is the computational time. Due to using two forward passes, the model requires more time to generate the results compared to the other models.

References

Mahtab Ahmed, Jumayel Islam, Muhammad Rifayat Samee, and Robert E Mercer. 2019a. Identifying protein-protein interaction using tree LSTM and structured attention. In *2019 IEEE 13th Int. Conf. on Semantic Computing (ICSC)*, pages 224–231.

Mahtab Ahmed, Muhammad Rifayat Samee, and Robert E Mercer. 2019b. Improving tree-LSTM with tree attention. In *2019 IEEE 13th Int. Conf. on Semantic Computing (ICSC)*, pages 247–254.

Mahtab Ahmed, Muhammad Rifayat Samee, and Robert E Mercer. 2019c. You only need attention to traverse trees. In *Proc. 57th Ann. Meet. of the Assoc. for Computational Linguistics*, pages 316–322.

Antti Airola, Sampo Pyysalo, Jari Björne, Tapio Pahikkala, Filip Ginter, and Tapio Salakoski. 2008. All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC Bioinformatics*, 9 Suppl 11:S2.

Melina Altmann, Stefan Altmann, Patricia A Rodriguez, Benjamin Weller, Lena Elorduy Vergara, Julius Palme, Nora Marín-de la Rosa, Mayra Sauer, Marion Wenig, José Antonio Villaécija-Aguilar, et al. 2020. Extensive signal integration by the phytohormone protein network. *Nature*, 583(7815):271–276.

Masaki Asada, Makoto Miwa, and Yutaka Sasaki. 2021. Using drug descriptions and molecular structures for drug–drug interaction extraction from literature. *Bioinformatics*, 37(12):1739–1746.

Masaki Asada, Makoto Miwa, and Yutaka Sasaki. 2023. Integrating heterogeneous knowledge graphs into drug–drug interaction extraction from the literature. *Bioinformatics*, 39(1):btac754.

William A Baumgartner, Zhiyong Lu, Helen L Johnson, J Gregory Caporaso, Jesse Paquette, Anna Lindemann, Elizabeth K White, Olga Medvedeva, K Bretonnel Cohen, and Lawrence Hunter. 2008. Concept recognition for extracting protein interaction relations from biomedical text. *Genome biology*, 9(2):1–15.

Razvan Bunescu, Ruifang Ge, Rohit Kate, Edward Marcotte, Raymond Mooney, Arun Ramani, and Yuk Wah Wong. 2005. Comparative experiments on learning information extractors for proteins and their interactions. *Art. Intell. in Medicine*, 33(2):139–155.

Yung-Chun Chang, Chun-Han Chu, Yu-Chen Su, Chien Chin Chen, and Wen-Lian Hsu. 2016. Pipe: a protein–protein interaction passage extraction module for biocreative challenge. *Database*, 2016.

Sung-Pil Choi. 2016. Extraction of protein-protein interactions (ppis) from the literature by deep convolutional neural networks with various feature embeddings. *Journal of Information Science*, 44.

Jing Ding, Daniel Berleant, Dan Nettleton, and Eve Wurtele. 2001. Mining medline: abstracts, sentences, or phrases? In *Biocomputing 2002*, pages 326–337. World Scientific.

Hao Fei, Yue Zhang, Yafeng Ren, and Donghong Ji. 2021. A span-graph neural model for overlapping entity relation extraction in biomedical texts. *Bioinformatics*, 37(11):1581–1589.

Katrin Fundel, Robert Küffner, and Ralf Zimmer. 2007. Relex—relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.

David E Gordon, Gwendolyn M Jang, Mehdi Bouhaddou, Jiewei Xu, Kirsten Obernier, Kris M White, Matthew J O'Meara, Veronica V Rezelj, Jeffrey Z Guo, Danielle L Swaney, et al. 2020. A sars-cov-2 protein interaction map reveals targets for drug repurposing. *Nature*, 583(7816):459–468.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360.

OM Zack Howard, Hui Fang Dong, Aiko-Konno Shirakawa, and Joost J Oppenheim. 2000. LEC induces chemotaxis and adhesion by interacting with CCR1 and CCR8. *Blood, The Journal of the American Society of Hematology*, 96(3):840–845.

Yu-Lun Hsieh, Yung-Chun Chang, Nai-Wen Chang, and Wen-Lian Hsu. 2017. Identifying protein-protein interactions in biomedical literature using recurrent neural networks with long short-term memory. In *Proc. of the Eighth Int. Joint Conf. on Natural Language Proc. (Vol. 2: Short Papers)*, pages 240–245.

Lei Hua and Chanqin Quan. 2016. A shortest dependency path based convolutional neural network for protein-protein relation extraction. *BioMed Research International*, 2016.

Martin Krallinger, Florian Leitner, Carlos Rodriguez-Penagos, and Alfonso Valencia. 2008. Overview of the protein-protein interaction annotation extraction task of BioCreative II. *Genome Biology*, 9(2):1–19.

Jiwei Li, Thang Luong, Dan Jurafsky, and Eduard Hovy. 2015. When are tree structures necessary for deep learning of representations? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2304–2314.

Shengyu Liu, Buzhou Tang, Qingcai Chen, and Xiaolong Wang. 2016. Drug-drug interaction extraction via convolutional neural networks. *Computational and Mathematical Methods in Medicine*, 2016.

Agnes Lydia and Sagayaraj Francis. 2019. Adagrad—an optimizer for stochastic gradient descent. *Int. J. Inf. Comput. Sci*, 6(5):566–568.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Gurusamy Murugesan, Sabenabanu Abdulkadhar, and Jeyakumar Natarajan. 2017. Distributed smoothed tree kernel for protein-protein interaction extraction from the biomedical literature. *PLOS ONE*, 12:e0187379.

Ernst H Nauta, Herman Mattie, and Wim RO Goslings. 1974. Effect of probenecid on the apparent volume of distribution and elimination of cloxacillin. *Antimicrobial Agents and Chemotherapy*, 6(3):300–303.

Claire Nédellec. 2005. Learning language in logic-genic interaction extraction challenge. In *Proceedings of the Learning Language in Logic 2005 Workshop (LLL05)*, pages 31–37.

Yifan Peng, Chih-Hsuan Wei, and Zhiyong Lu. 2016. Improving chemical disease relation extraction with rich features and weakly labeled data. *Journal of Cheminformatics*, 8(1):1–12.

Long N Phan, James T Anibal, Hieu Tran, Shaurya Chanana, Erol Bahadroglu, Alec Peltekian, and Grégoire Altan-Bonnet. 2021. SciFive: a text-to-text transformer model for biomedical literature. *arXiv preprint arXiv:2106.03598*.

Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Björne, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski. 2007. Bioinfer: a corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(1):1–24.

A David Rodrigues. 2019. *Drug-Drug Interactions*, 2nd edition. CRC Press.

David L Sackett. 1997. Evidence-based medicine. *Seminars in Perinatology*, 21(1):3–5.

Sunil Kumar Sahu and Ashish Anand. 2018. Drug-drug interaction extraction from biomedical texts using long short-term memory network. *Journal of Biomedical Informatics*, 86:15–24.

Isabel Segura-Bedmar, Paloma Martínez Fernández, and María Herrero Zazo. 2013. SemEval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 341–350.

Sudipta Singha Roy and Robert E Mercer. 2022. Protein-protein interaction extraction using attention-based tree-structured neural network models. In *The International FLAIRS Conference Proceedings*, volume 35.

Sudipta Singha Roy and Robert E. Mercer. 2023. Identifying protein-protein interaction using tree-transformers and heterogeneous graph neural network. *The International FLAIRS Conference Proceedings*, 36(1).

Samuel Sledzieski, Rohit Singh, Lenore Cowen, and Bonnie Berger. 2021. D-script translates genome to phenome with sequence-based, structure-aware, genome-scale predictions of protein-protein interactions. *Cell Systems*, 12(10):969–982.e6.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *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)*, pages 1556–1566.

Zhan Tang, Xuchao Guo, Zhao Bai, Lei Diao, Shuhan Lu, and Lin Li. 2022. A protein-protein interaction extraction approach based on large pre-trained language model and adversarial training. *KSII Transactions on Internet and Information Systems (TIIS)*, 16(3):771–791.

Domonkos Tikk, Philippe Thomas, Peter Palaga, Jörg Hakenberg, and Ulf Leser. 2010. A comprehensive benchmark of kernel methods to extract protein–protein interactions from literature. *PLoS Computational Biology*, 6(7):e1000837.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219.

Yaushian Wang, Hung-Yi Lee, and Yun-Nung Chen. 2019. Tree transformer: Integrating tree structures into self-attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1061–1070.

Shweta Yadav, Asif Ekbal, Sriparna Saha, Ankit Kumar, and Pushpak Bhattacharyya. 2019. Feature assisted stacked attentive shortest dependency path based Bi-LSTM model for protein–protein interaction. *Knowledge-Based Systems*, 166:18–29.

Shweta Yadav, Srivastsa Ramesh, Sriparna Saha, and Asif Ekbal. 2020. Relation extraction from biomedical and clinical text: Unified multitask learning framework. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 19(2):1105–1116.

Yu Yao, Xiuquan Du, Yanyu Diao, and Huaixu Zhu. 2019. An integration of deep learning with feature embedding for protein–protein interaction prediction. *PeerJ*, 7:e7126.

Kaixian Yu, Pei-Yau Lung, Tingting Zhao, Peixiang Zhao, Yan-Yuan Tseng, and Jinfeng Zhang. 2018. Automatic extraction of protein-protein interactions using grammatical relationship graph. *BMC Medical Informatics and Decision Making*, 18:35–43.

Yijia Zhang, Hongfei Lin, Zhihao Yang, Jian Wang, Shaowu Zhang, Yuanyuan Sun, and Liang Yang. 2018. A hybrid model based on neural networks for biomedical relation extraction. *Journal of Biomedical Informatics*, 81:83–92.

Zhehuan Zhao, Zhihao Yang, Hongfei Lin, Jian Wang, and Song Gao. 2016. A protein-protein interaction extraction approach based on deep neural network. *International Journal of Data Mining and Bioinformatics*, 15(2):145–164.