Biomedical Event Extraction Based on Knowledge-driven Tree-LSTM

Diya Li[†], **Lifu Huang**[†], **Heng Ji**[†], **Jiawei Han**[*]
[†]Computer Science Department, Rensselaer Polytechnic Institute
{lid18,huangl7,jih}@rpi.edu
[*]Computer Science Department, University of Illinois at Urbana Champaign
hanj@illinois.edu

Abstract

Event extraction for the biomedical domain is more challenging than that in the general news domain since it requires broader acquisition of domain-specific knowledge and deeper understanding of complex contexts. To better encode contextual information and external background knowledge, we propose a novel knowledge base (KB)-driven tree-structured long short-term memory networks (Tree-LSTM) framework, incorporating two new types of features: (1) dependency structures to capture wide contexts; (2) entity properties (types and category descriptions) from external ontologies via entity linking. We evaluate our approach on the BioNLP shared task with Genia dataset and achieve a new state-of-the-art result. In addition, both quantitative and qualitative studies demonstrate the advancement of the Tree-LSTM and the external knowledge representation for biomedical event extraction.

1 Introduction

Biomedical information extraction is widely used to assist the biology community on knowledge acquisition and ontology construction. Biomedical events generally refer to a change of status, particularly on proteins or genes. The goal of event extraction is to identify triggers and their arguments from biomedical text, and then assign an event type to each trigger and a role to each argument. For example, in the sentence shown in Figure 1, it includes a **gene expression** and a **positive regulation** event mention, both triggered by the word *transduced*. *Tax* is the **Theme** argument of the **gene expression** event. An event could also serve as an argument of another event, leading to a nested structure. For instance, the **gene expression** event triggered by *transduced* is also a **Theme** argument of the **positive regulation** event as shown in Figure 1.

Earlier studies on biomedical event extraction rely on kernel classification methods like the support vector machines (SVMs) (Björne and Salakoski, 2011; Venugopal et al., 2014) using hand-crafted features, which require high engineering effort and domain-specific knowledge. Recent distributional representation based approaches (Rao et al., 2017; Björne and Salakoski, 2018) explore deep neural networks which only require distributed semantic features. However, different from event extraction in the general news domain, biomedical event extraction requires broad acquisition of domain-specific knowledge and deep understanding of complex contexts. For example, in Genia event extracton of BioNLP shared task 2011 (Kim et al., 2011), about 80% of entity mentions are abbreviations of genes, proteins and diseases while more than 36% of event triggers and arguments are separated with more than 10 words.

In order to efficiently capture indicative information from broad contexts, we first adopt tree structure based long short-term memory (Tree-LSTM) networks. Compared to the linear chain structured LSTM, the Tree-LSTM takes tree-structured network topology into consideration. As shown in the top frame of Figure 1, Tree-LSTM takes the dependency tree structure of each sentence as input and gradually incorporates the information from the whole subtree into each node. Dependency tree structure can connect semantically related concepts, and thus shorten the distance between a trigger and its arguments significantly. For instance, in the following sentence "*... , which* **binds** *to the enhancer A located in the promoter of the mouse MHC class I gene* **H-2Kb**, ...", when determining the trigger type of *binds*, we need to carefully select its contextual words, such as *H-2Kb*, which indicates the object of *binds*. However, *binds* and *H-2Kb* are sepa-

Proceedings of NAACL-HLT 2019, pages 1421–1430
Minneapolis, Minnesota, June 2 - June 7, 2019.

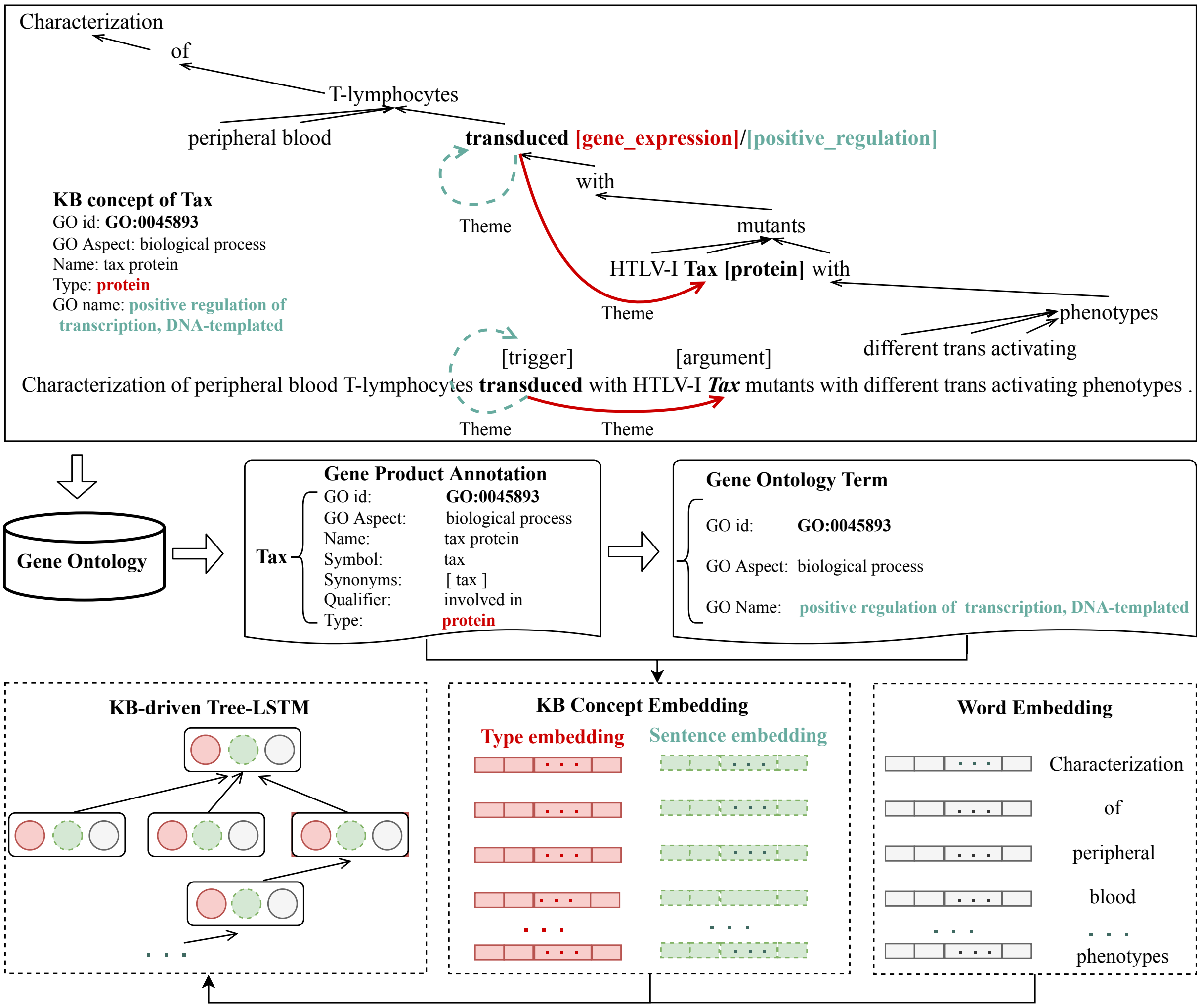


Figure 1: The framework of the KB-driven Tree-LSTM model. The upper frame shows the dependency tree structure and event annotations of a sentence; the middle frame demonstrates the knowledge base information obtained from the Gene Ontology for *Tax*; the bottom frame describes the KB-driven Tree-LSTM which takes the KB concept embedding and word embedding as input.

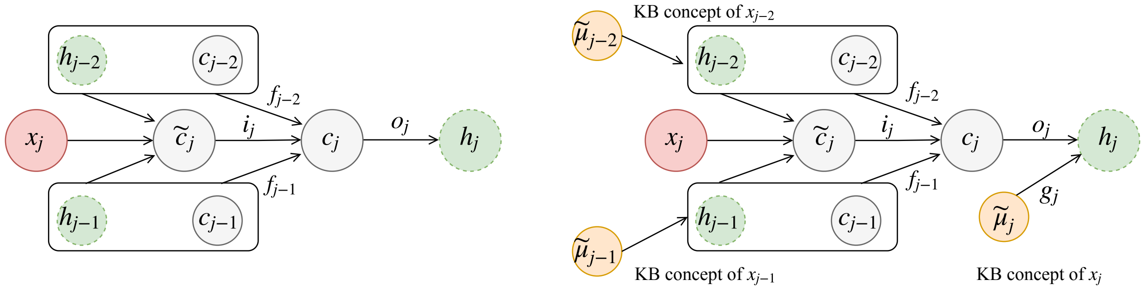
rated with 16 words which is difficult for a chain-structured LSTM to capture their long distance dependency, while within dependency tree structure, their distance is significantly shortened to 7.

Moreover, to better capture domain-specific knowledge, we further propose to leverage the external knowledge bases (KBs) to acquire properties of all the biomedical entities. The KB properties are extremely beneficial for our model to learn patterns more explicitly. Take the entity *Tax* in Figure 1 as an example, it's a protein often involved in the biological process of *positive regulation of transcription* referred to Gene Ontology (Ashburner et al., 2000). This function description provides crucial clues to determine the type of *transduced* as **positive regulation**. Therefore, to capture such knowledge from external KBs, for each entity, we first learn a KB concept embedding from its properties, and then automatically incorporate the KB representation into its Tree-LSTM hidden state with a gate function.

Our contributions are twofold: First, to the best of our knowledge, it's the first time to adopt Tree-LSTM for biomedical event extraction to effectively capture the wide contexts. Second, we further incorporate external knowledge from domain-specific KBs into the Tree-LSTM, which yields state-of-the-art performance on Genia event extraction shared task.

2 KB-driven Tree-LSTM for Event Extraction

In this section, we present our KB-driven Tree-LSTM approach for biomedical event extraction.



A. a Tree-LSTM unit

B. a KB-driven Tree-LSTM unit

Figure 2: (A): a Tree-LSTM unit. (B): a KB-driven Tree-LSTM unit. The yellow circles with $\widetilde{\mu}$ notations denote external KB concept embeddings.

We first introduce the Tree-LSTM framework, and then describe the construction of KB concept embedding for each entity. Finally we incorporate the KB concept embedding into a Tree-LSTM and apply it for event trigger and argument extraction.

2.1 Tree-LSTM

The Tree-LSTM (Tai et al., 2015) is a variation of LSTM (Hochreiter and Schmidhuber, 1997) to a tree-structured network topology. It shows improvement in representing sentence semantic meaning compared to sequential LSTM such as Bidirectional LSTM (BiLSTM) (Graves et al., 2013). The main difference between sequential LSTM and Tree-LSTM is, at each time step, the former calculates its hidden state from the input at the current time step and the hidden state from previous step, while Tree-LSTM computes its hidden state from the input token and the hidden states of all its children nodes from the tree structure. A Tree-LSTM reduces to sequential LSTM when each node in the tree only has one child. Figure 2 (A) shows a Tree-LSTM unit. In order to obtain the hidden state h_j of an input token x_j, the unit calculates all of its children hidden states (h_{j-1}, h_{j-2}) through depth-first traversal.

2.2 Constructing KB Concept Embedding

For the biomedical event extraction, we mainly explore the Gene Ontology as our external KB since it provides detailed descriptions for each gene and gene product attributes across all species. It consists of two types of information: (1) the gene ontology (GO) defines all the gene functions, relations between these gene functions, and aspects used to describe the gene functions, including molecular function, cellular component and biological process. (2) the gene product annotations (GO Anno) provide all entity related attributes, such as the full entity name, entity type, as well as the gene functions it is related to. For example, in Figure 1, given the entity *tax*, from the gene product annotations, we can get its full entity name as *tax protein* which is a type of *proteins* and it's related to a function about *biological process*. From the gene ontology, we can further determine the specific function that *tax* is related to *positive regulation of transcription* in terms of *biological process* aspect.

In order to leverage the external KB information, we first apply QuickGO API (Binns et al., 2009) to link each entity mention to the Gene Ontology and retrieve all the KB annotations. For each entity, we carefully select two types of properties which are beneficial for event extraction task: the entity type (e.g., *protein* for *tax*) and the gene ontology function it is related to (e.g., *positive regulation of transcription* for *tax*). The entity type can facilitate the explicit pattern learning for argument role labeling, for example, the gene expression event pattern (Theme: Protein, Trigger: transduced) is more popular than (Theme: Tax, Trigger: transduced) in Figure 1. The gene ontology function can provide implicit clues to determine the trigger type as aforementioned in Section 1.

As shown in Figure 1, we assign a word embedding which pretrained on PubMed and PMC texts (Moen and Ananiadou, 2013) to represent each entity type. For each gene ontology function which is usually a long phrase, we use a state-of-the-art sentence embedding approach (Conneau et al., 2017) to automatically learn a vector representation. We then concatenate these two types of KB property representations as the final KB concept embedding.

2.3 Event Trigger Extraction

After obtaining the KB concept embeddings, we further incorporate them into the Tree-LSTM to leverage the domain-specific knowledge.

Given a sentence, for example the sentence shown in Figure 3, we first perform the dependency parsing with the Stanford dependency parser (Chen and Manning) and obtain a dependency tree structure. For each node j in the tree structure, $C(j)$ is the set of children nodes of node j and μ_k is the KB concept embedding of node k. We set μ_k to 0 if node k is not a biomedical entity. $\widetilde{\mu}_j$ denotes the sum of the KB concept embeddings of j's children nodes and $\widetilde{h}_j$ is the sum of the hidden states of j's children nodes:

$$\widetilde{h}_j = \sum_{k \in C(j)} h_k$$
$$\widetilde{\mu}_j = \sum_{k \in C(j)} \mu_k$$

where h_k is the hidden state of node k.

Then we incorporate the KB concept embeddings into the input, forget, and output gates of the Tree-LSTM:

$$\begin{aligned} i_j &= \sigma(W_i[x_j, \widetilde{h}_j, \widetilde{\mu}_j] + b_i) \\ f_{jk} &= \sigma(W_f[x_j, h_k, \widetilde{\mu}_k] + b_f) \\ o_j &= \sigma(W_o[x_j, \widetilde{h}_j, \widetilde{\mu}_j] + b_o) \end{aligned}$$

where i_j and o_j are the input gate and the output gate for node j respectively. f_{jk} is the forget gate for node j in terms of its child node k. W_i, W_f, and W_o are learnable parameters, b_i, b_f and b_o are bias terms. Thus, for each node j, the input gate gathers all KB information from its children nodes, and the output gate balances the meaningful information from its local contexts and the KB concept embeddings of its children nodes.

Besides adding the KB concept embeddings into the three gates to select useful KB formation implicitly, similar to Ma et al. (2018), we also introduce a knowledge specific output gate g_j to explicitly incorporate knowledge information into each node's hidden state. While different from Ma et al. (2018) which only considers the knowledge concept embedding of each node itself, we use the sum of the KB concept embeddings of the whole subtree instead:

$$g_j = \sigma(W_g[x_j, \widetilde{h}_j, \widetilde{\mu}_j] + b_g)$$

where W_g is a weight matrix to be learned, b_g is the bias term.

As demonstrated in Figure 2 (B), we eventually combine the implicit way of incorporating KB information into the input, output and forget gates and an explicit way of directly incorporating the KB information into a node's hidden state:

$$\begin{aligned} \widetilde{c}_j &= \tanh(W_c[x_j, \widetilde{h}_j] + b_c) \\ c_j &= \sum_{k \in C(j)} f_{jk} \odot c_k + i_j \odot \widetilde{c}_j \\ h_j &= o_j \odot \tanh(c_j) + g_j \odot \tanh(W_\mu \widetilde{\mu}_j) \end{aligned}$$

where c_j is the memory cell, W_c and W_μ are weight matrices to be learned.

After getting the hidden state h_j of each node j, we use a softmax classifier to predict a label for each node, and optimize the parameters by minimizing a negative log-likelihood loss.

2.4 Event Argument Role Labeling

After detecting all candidate triggers, we further extract arguments for each trigger. The Genia event extraction shared task provides the annotations of all entity mentions. Thus, for each trigger, we use all the entity mentions that occur in the same sentence as its candidate arguments, and then assign an argument role or *None*. Different from trigger extraction, we use the shortest dependency path (SDP) within the dependency tree structure instead of the surface contexts to better capture the dependency between the trigger and each argument.

Taking the sentence in Figure 3 as an example, given a trigger *transcription* and a candidate argument *OBF-1*, we first perform dependency parsing and extract the shortest dependency path between *transcription* and *OBF-1* with the Dijkstra's algorithm (Johnson, 1973) and obtain the shortest dependency path *transcription* → *of* → *genes* → *OBF-1*. We use the same KB-driven Tree-LSTM architecture as introduced in Section 2.3 to encode each node into a new hidden state representation. We use the hidden state of the root node h_0 as the overall vector representation of the whole dependency path. Finally, we feed the concatenation of h_0 with the hidden state of the trigger and argument as input to another softmax to predict the argument role. We also optimize the model by minimizing a negative log-likelihood loss.

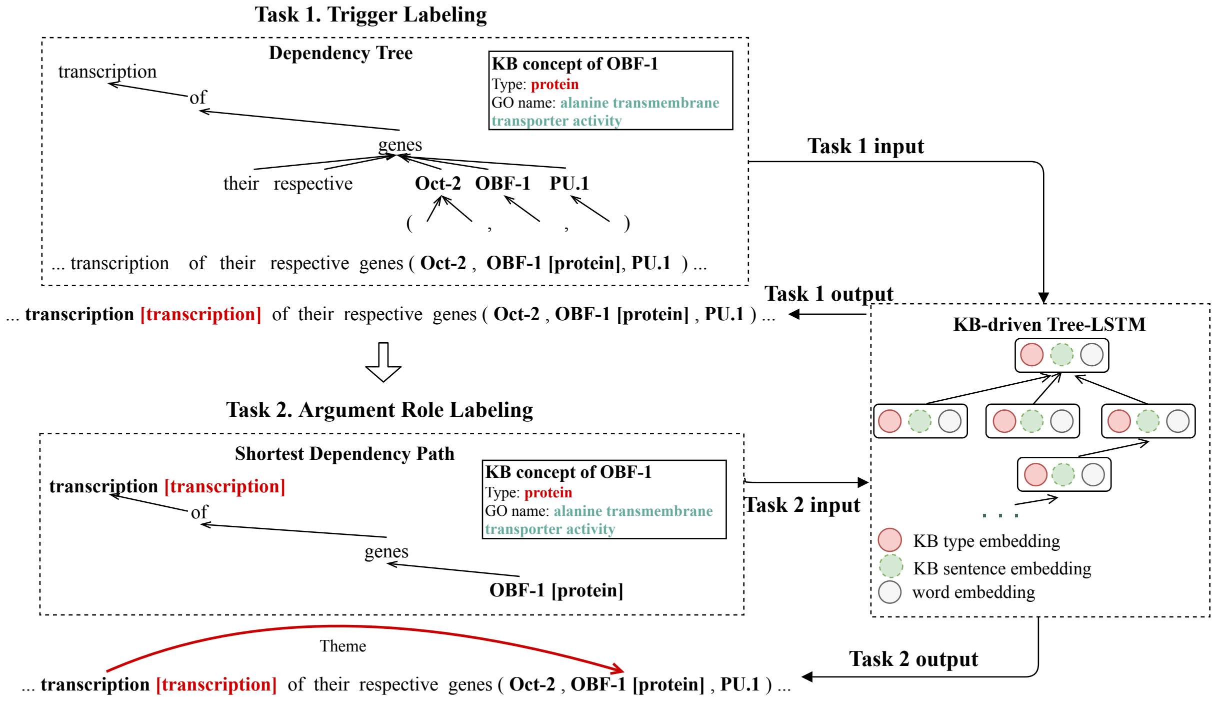

Task 1. Trigger Labeling
Dependency Tree
transcription
of
genes
their respective
Oct-2 OBF-1 PU.1
(, ,)
KB concept of OBF-1
Type: protein
GO name: alanine transmembrane transporter activity
... transcription of their respective genes (Oct-2 , OBF-1 [protein], PU.1) ...
Task 1 input
Task 1 output
... transcription [transcription] of their respective genes (Oct-2 , OBF-1 [protein] , PU.1) ...
KB-driven Tree-LSTM
Task 2. Argument Role Labeling
Shortest Dependency Path
transcription [transcription]
of
genes
OBF-1 [protein]
KB concept of OBF-1
Type: protein
GO name: alanine transmembrane transporter activity
Task 2 input
KB type embedding
KB sentence embedding
word embedding
Task 2 output
Theme
... transcription [transcription] of their respective genes (Oct-2 , OBF-1 [protein] , PU.1) ...

Figure 3: Examples of trigger labeling and argument role labeling via a KB-driven Tree-LSTM.

Event Type	Core Arguments
Gene expression	Theme(P)
Transcription	Theme(P)
Protein catabolism	Theme(P)
Phosphorylation	Theme(P)
Localization	Theme(P)
Binding	Theme(P)+
Regulation	Theme(P/E), Cause(P/E)
Positive regulation	Theme(P/E), Cause(P/E)
Negative regulation	Theme(P/E), Cause(P/E)
Genia corpus 2011 statistics	
events	14496
sentences	11581
nested events	37.2%
intersentence events	6.0%
abbrev. of entities	15912

Table 1: Predefined event types with accepted argument roles in Genia event extraction task, and data statistics of Genia event extraction 2011 dataset. P: protein; E: event.

3 Experiment

3.1 Task Description

The Genia Event Extraction task is the main task in the BioNLP Shared Task series (Kim et al., 2009, 2011; Nédellec et al., 2013). The Genia task defines 9 fine-grained event types as shown in Table 1. Note that a **Binding** event may take more than one protein as its **Theme** arguments. A **Regulation** event may take one protein or event as its **Theme** argument and also optionally take one protein or event as its **Cause** argument. A **Regulation** event taking an event as its argument will lead to a nested structure. 37.2% nested events are observed in Genia 2011 corpus (Björne and Salakoski, 2011). There are 6.0% inter-sentence events while our model only focuses on sentence-level event extraction.

3.2 Experimental Setup

We apply our KB-driven Tree-LSTM model on Genia 2011 data set. The entities in Genia data set are manually annotated and given as part of the input.

We evaluate our results on the test set using the official online tool provided by the Genia task organizers.[1] Following previous studies (Björne and Salakoski, 2011; Venugopal et al., 2014; Rao et al., 2017; Björne and Salakoski, 2018), we report scores obtained by the approximate span (allowing trigger spans to differ from gold spans by single words). As we only focus on matching core arguments, we use recursive matching criterion for evaluation which not requires matching of additional arguments for events referred from other events (Kim et al., 2011).

We use the word embedding pretrained on PubMed and PMC texts (Moen and Ananiadou,

[1] http://bionlp-st.dbcls.jp/GE/2011/eval-test/

2013) for word and type embeddings. The hyper-parameters are tuned on the development set and listed in Table 2. Word representations are updated during training with an initial learning rate of 0.1.

Parameter	Value
Word embedding size	200
Type embedding size	200
Sentence embedding size	4096
Tree-LSTM hidden size	100
Batch size	25
Epoch size	30
Dropout rate	0.5
Learning rate	0.05
Initial embedding learning rate	0.1
Optimizer	AdaGrad

Table 2: Hyper-parameters.

3.3 Results and Error Analysis

Table 3 shows the final event extraction results of applying our KB-driven Tree-LSTM model on Genia 2011 dataset with the comparison of only using Tree-LSTM and a standard BiLSTM model. Tree-LSTM outperforms the BiLSTM baseline which indicates the power of Tree-LSTM in dealing with long-distance dependency structure in biomedical literature. By incorporating external KB information, our approach achieves about 2.12% F-score gain comparing to Tree-LSTM, which demonstrates the effectiveness of the KB properties for biomedical event extraction. We will show detailed analysis in Section 3.4.

Table 4 presents the previous event extraction results from the BioNLP shared task using the same corpus. Our approach outperforms all previous methods. Among them, the systems TEES (Björne and Salakoski, 2011), EventMine-CR (Miwa et al., 2012) and Stacked Generalization (Majumder et al., 2016) are based on SVMs with well designed features. FAUST (Riedel and McCallum, 2011) and BioMLN (Venugopal et al., 2014) use jointed inference models. Björne and Salakoski (2018) adopts a convolutional neural networks (CNNs) with abundant features derived from TEES system. In our work, instead of using high-dimensional features with manual effort as in these previous models, our approach only requires pretrained distributed word representations as input features.

System	Event Type	Rec	Prec	F1
KB-driven Tree-LSTM	Gene expression	74.35	87.24	80.28
	Transcription	69.54	82.31	75.39
	Protein catabolism	46.67	87.50	60.87
	Phosphorylation	81.62	87.28	84.36
	Localization	59.69	80.28	68.47
	Simple total	72.62	85.95	78.73
	Binding	37.68	53.16	44.10
	Regulation	36.62	53.61	43.52
	Positive regulation	41.37	57.90	48.26
	Negative regulation	46.06	52.39	49.02
	Regulation total	41.73	55.73	47.72
	Event total	**52.14**	**67.01**	**58.65**
Tree-LSTM	**Simple total**	71.22	83.41	76.83
	Binding	34.83	48.72	40.62
	Regulation total	39.78	53.54	45.64
	Event total	**50.28**	**64.56**	**56.53**
BiLSTM	**Simple total**	68.09	78.75	73.03
	Binding	38.49	43.05	40.65
	Regulation total	37.64	53.81	44.30
	Event total	**48.44**	**62.18**	**54.46**

Table 3: Precision (Prec), recall (Rec) and F-score (F1) results achieved by the KB-driven Tree-LSTM model on the test set of BioNLP Genia 2011, evaluated on approximate span and recursive criteria.

System	Rec	Prec	F1
TEES(Björne and Salakoski, 2011)	49.56	57.65	53.30
FAUST(Riedel and McCallum, 2011)	49.41	64.75	56.04
EventMine-CR(Miwa et al., 2012)	53.35	63.48	57.98
BioMLN(Venugopal et al., 2014)	53.42	63.61	**58.07**
Stacked Generalization(Majumder et al., 2016)	48.96	66.46	56.38
CNN(Björne and Salakoski, 2018)	49.94	69.45	**58.07**

Table 4: State-of-the-art system results evaluated on BioNLP Genia 2011 test dataset with approximate span and recursive criteria.

We notice that our approach achieves high scores on **Simple** event types but get relatively low scores on **Binding** event and **Regulation** event types. We analyze the results and find that **Binding** event extraction is more challenging since it usually has multiple arguments. For example, Figure 4 shows two sentences which are chosen from the output of the development data set. There are two **Binding** event mentions in the first sentence: E1 (Trigger: interacting, Type: Binding, Theme: RUNX1, Theme2: p3000) and E2 (Trigger: binding, Type: Binding, Theme: CREB). Our model mistakenly extracts *CREB* as a *Theme* of E1 since *CREB* is highly related to protein p300 in the dependency tree structure.

Regulation events are considered as the most challenging event type because they usually have an optional **Cause** argument and are involved in nested structures, which are not handled well by most of current event extraction approaches. In addition, intuitively, most trigger words are verbs or nouns. We rank all the trigger words in the train-

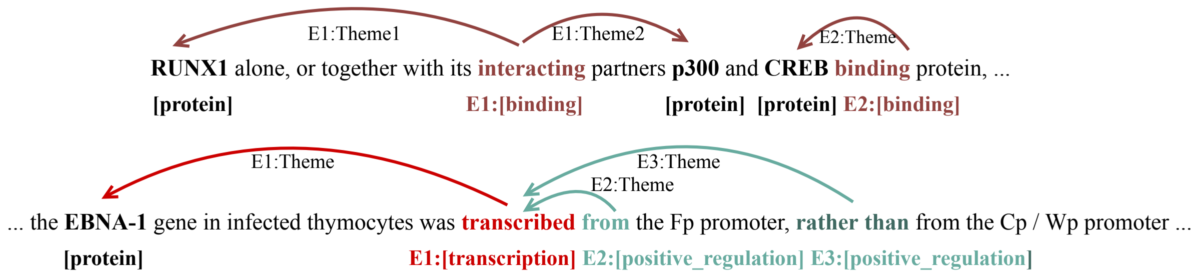

E1:Theme1
E1:Theme2
E2:Theme
RUNX1 alone, or together with its interacting partners p300 and CREB binding protein, ...
[protein]
E1:[binding]
[protein] [protein] E2:[binding]
E1:Theme
E3:Theme
E2:Theme
... the EBNA-1 gene in infected thymocytes was transcribed from the Fp promoter, rather than from the Cp / Wp promoter ...
[protein]
E1:[transcription] E2:[positive_regulation] E3:[positive_regulation]

Figure 4: Case study on binding event and regulation event types.

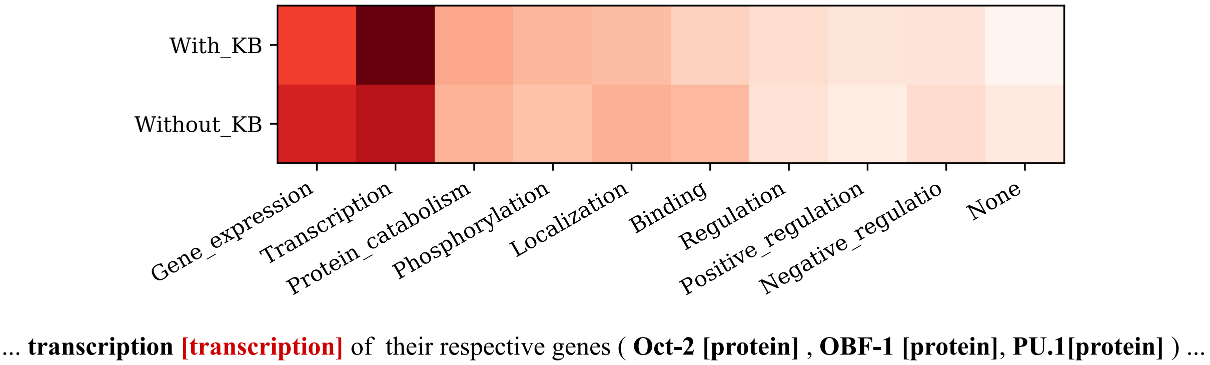


Figure 5: Visualization of the effect of KB concept embeddings on trigger labeling for the word *transcription*.

ing data set according to their frequency, and find that most of spurious errors for **Regulation** event trigger extraction occur when the trigger words are prepositions or conjunctions. For instance, in Figure 4, the second sentence contains two positive **Regulation** events triggered by a preposition *from* and a conjunction *rather than*. Such function words are rarely annotated as triggers and our KB-aware Tree-LSTM cannot well collect meaningful contexts from their subtrees.

3.4 Effect of KB concepts

As shown in Table 3, we achieve about 3.5% and 2.1% F1 score gain on **Binding** and **Regulation** event types by leveraging external KB information into the Tree-LSTM. In order to show the effect of KB concept embeddings, we visualize the probabilities of word *transcription* to be predicted for each event type. As Figure 5 shows, by adding KB concept embeddings, the function description *positive regulation of transcription, DNA-templated* provided by the biomedical entity *OBF-1* significantly enhances the probability of *transcription* being predicted to a *Transcription* event type.

Similarly, Figure 6 visualizes the probabilities of the E1 event mention (Trigger: transduced, Type: gene_expression, Theme: Tax) to be predicted as an argument of E2 event mention (Trigger: transduced, Type: positive regulation, Theme: E1). We can see that, without using KB information, the Tree-LSTM mistakenly predict the argument role of E1 as **None**. In contrast, by incorporating KB concept embeddings, especially

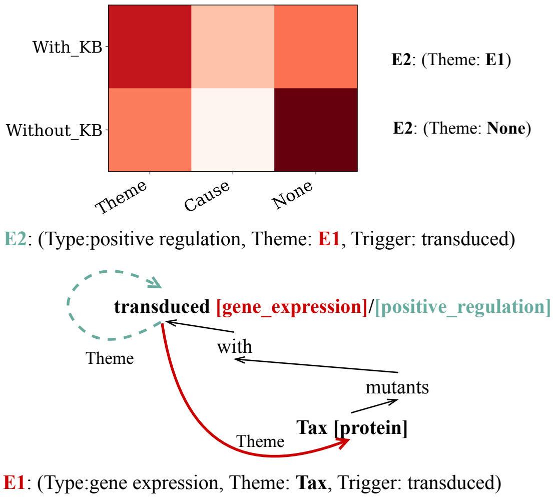


Figure 6: Visualization of the effect of KB concept embedding on argument role labeling for a Positive Regulation event triggered by *transduced* and a Gene Expression event E1 (Theme: Tax, Trigger: transduced).

the information from the function description *positive regulation of transcription, DNA-templated* for *Tax*, our approach successfully promotes the probability of E1 being predicted as the *Theme* of E2.

4 Related Work

As a crucial task in information extraction, event extraction has gained a lot of interest. In general news domain, previous work on event extraction can be divided into two main categories. The first is feature-based methods which mainly focus on feature design, leveraging local features (Grishman et al., 2005; Ahn, 2006) and global features (Ji and Grishman, 2008; Liao and Grishman, 2011; Huang and Riloff, 2012) to improve the performance. Some studies proposed joint models to overcome the error propagation problem (Poon and Vanderwende, 2010; Riedel et al., 2009; Li et al., 2013; Venugopal et al., 2014; Li et al., 2014). The second category includes distributional representation based methods which have been applied into event extraction extensively. Most of these approaches are based on the standard Convolutional Neural Networks (CNNs) (Chen et al., 2015; Nguyen and Grishman, 2015, 2016), Recurrent Neural Networks (RNNs) (Nguyen et al., 2016), generative adversarial networks (Hong et al., 2018), zero-shot learning (Huang et al., 2017) and advanced attention mechanisms (Liu et al., 2018b; Chen et al., 2018).

Our work is also related to the studies which leverage the external knowledge base for information extraction. Liu et al. (2017) takes advantage of external resources, such as FrameNet, to label events while Chen et al. (2017) adopts distance supervision to augment the training data. Liu et al. (2018a) develops an attention-based model for event extraction. What's more, shortest dependency path is broadly explored for information extraction, especially for relation classification (Xu et al., 2015; Miwa and Bansal, 2016) and shows promising benefits.

Biomedical event extraction task part of the BioNLP Shared Task series (Kim et al., 2009, 2011; Nédellec et al., 2013). Previous studies mainly explore local and global features with SVM model (Miwa et al., 2010, 2012; Björne and Salakoski, 2013; Majumder et al., 2016). Riedel and McCallum (2011) develop a joint model with dual decomposition. Cohen et al. (2009), Kilicoglu and Bergler (2011) and Bui et al. (2013) develop rule-based methods and achieve high precision. Venugopal et al. (2014) leverage Markov logic networks for joint inference. Rao et al. (2017) uses the Abstract Meaning Representations (AMR) to extract events based on the assumption that an event structure can be derived from an AMR subgraph. Recently, some representation-based models (Jagannatha and Yu, 2016; Rao et al., 2017; Björne and Salakoski, 2018) have been proposed while most of them adopt the widely used CNNs and RNNs with features derived from the biomedical text. Lim et al. (2018) implements a binary Tree-LSTM architecture for biomedical relation extraction. Compared with these methods, our approach only requires pre-trained distributed word representations as input features and incorporates meaningful KB information into a Tree-LSTM.

5 Conclusions and Future Work

In this paper, we show the effectiveness of using a KB-driven tree-structured LSTM for event extraction in biomedical domain. The Tree-LSTM can efficiently capture semantically related concepts for each node within the tree structure. By leveraging the external KB concept properties including the entity type and the function description, our approach is able to perform deep understanding of domain-specific expressions and connections. Without using manually designed high-dimensional features, our approach significantly outperforms all previous methods. In the future, we plan to explore a broader range of properties from KB to facilitate biomedical information extraction tasks.

Acknowledgments

This work was supported by the U.S. NSF No. 1741634, Air Force No. FA8650-17-C-7715 and ARL NS-CTA No. W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*.

Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. 2000. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25.

David Binns, Emily Dimmer, Rachael Huntley, Daniel Barrell, Claire O'donovan, and Rolf Apweiler. 2009. Quickgo: a web-based tool for gene ontology searching. *Bioinformatics*, 25(22):3045–3046.

Jari Björne and Tapio Salakoski. 2011. Generalizing biomedical event extraction. In *Proceedings of the BioNLP Shared Task 2011 Workshop*.

Jari Björne and Tapio Salakoski. 2013. Tees 2.1: Automated annotation scheme learning in the bionlp 2013 shared task. In *Proceedings of the BioNLP Shared Task 2013 Workshop*.

Jari Björne and Tapio Salakoski. 2018. Biomedical event extraction using convolutional neural networks and dependency parsing. In *Proceedings of the BioNLP 2018 workshop*.

Quoc-Chinh Bui, David Campos, Erik van Mulligen, and Jan Kors. 2013. A fast rule-based approach for biomedical event extraction. In *proceedings of the BioNLP shared task 2013 workshop*.

Danqi Chen and Christopher Manning. A fast and accurate dependency parser using neural networks. In *Proc. EMNLP2014*.

Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu, and Jun Zhao. 2017. Automatically labeled data generation for large scale event extraction. In *Proc. ACL2017*.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proc. ACL-IJCNLP2015*.

Yubo Chen, Hang Yang, Kang Liu, Jun Zhao, and Yantao Jia. 2018. Collective event detection via a hierarchical and bias tagging networks with gated multilevel attention mechanisms. In *Proc. EMNLP2018*.

K Bretonnel Cohen, Karin Verspoor, Helen L Johnson, Chris Roeder, Philip V Ogren, William A Baumgartner Jr, Elizabeth White, Hannah Tipney, and Lawrence Hunter. 2009. High-precision biological event extraction with a concept recognizer. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proc. EMNLP2017*.

Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. 2013. Hybrid speech recognition with deep bidirectional lstm. In *2013 IEEE workshop on automatic speech recognition and understanding*.

Ralph Grishman, David Westbrook, and Adam Meyers. 2005. NYU's English ACE 2005 system description. *ACE*, 5.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, pages 1735–1780.

Yu Hong, Wenxuan Zhou, jingli zhang jingli, Guodong Zhou, and Qiaoming Zhu. 2018. Self-regulation: Employing a generative adversarial network to improve event detection. In *Proc. ACL2018*.

Lifu Huang, Heng Ji, Kyunghyun Cho, and Clare R Voss. 2017. Zero-shot transfer learning for event extraction. *arXiv preprint arXiv:1707.01066*.

Ruihong Huang and Ellen Riloff. 2012. Bootstrapped training of event extraction classifiers. In *Proc. EACL2012*.

Abhyuday N Jagannatha and Hong Yu. 2016. Bidirectional rnn for medical event detection in electronic health records. In *Proc. NAACL-HLT2016*.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. *Proceedings of ACL-08: HLT*.

Donald B Johnson. 1973. A note on dijkstra's shortest path algorithm. *Journal of the ACM (JACM)*, 20(3):385–388.

Halil Kilicoglu and Sabine Bergler. 2011. Adapting a general semantic interpretation approach to biological event extraction. In *Proceedings of the BioNLP Shared Task 2011 Workshop*.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of bionlp'09 shared task on event extraction. In *Proceedings of the BioNLP Shared Task 2009 Workshop*.

Jin-Dong Kim, Yue Wang, Toshihisa Takagi, and Akinori Yonezawa. 2011. Overview of genia event task in bionlp shared task 2011. In *Proceedings of the BioNLP Shared Task 2011 Workshop*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proc. ACL2013*.

Qi Li, Heng Ji, Hong Yu, and Sujian Li. 2014. Constructing information networks using one single model. In *Proc. EMNLP2014*.

Shasha Liao and Ralph Grishman. 2011. Acquiring topic features to improve event extraction: in pre-selected and balanced collections. In *Proc. RANLP2011*.

Sangrak Lim, Kyubum Lee, and Jaewoo Kang. 2018. Drug drug interaction extraction from the literature using a recursive neural network. *PloS one*, 13(1):e0190926.

Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2018a. Event detection via gated multilingual attention mechanism. *Statistics*, 1000:1250.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *Proc. ACL2017*.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018b. Jointly multiple events extraction via attention-based graph information aggregation. In *Proc. EMNLP2018*.

Yukun Ma, Haiyun Peng, and Erik Cambria. 2018. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm. In *Proc. AAAI2018*.

Amit Majumder, Asif Ekbal, and Sudip Kumar Naskar. 2016. Biomolecular event extraction using a stacked generalization based classifier. In *Proc. ICON2016*.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770*.

Makoto Miwa, Rune Sætre, Jin-Dong Kim, and Jun'ichi Tsujii. 2010. Event extraction with complex event classification using rich features. *Journal of bioinformatics and computational biology*, 8(01):131–146.

Makoto Miwa, Paul Thompson, and Sophia Ananiadou. 2012. Boosting automatic event extraction from the literature using domain adaptation and coreference resolution. *Bioinformatics*, 28(13):1759–1765.

SPFGH Moen and Tapio Salakoski2 Sophia Ananiadou. 2013. Distributional semantics resources for biomedical text processing. In *Proc. LBM2013*.

Claire Nédellec, Robert Bossy, Jin-Dong Kim, Jung-Jae Kim, Tomoko Ohta, Sampo Pyysalo, and Pierre Zweigenbaum. 2013. Overview of bionlp shared task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop*.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proc. NAACL-HLT2016*.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proc. ACL-IJCNLP2015*.

Thien Huu Nguyen and Ralph Grishman. 2016. Modeling skip-grams for event detection with convolutional neural networks. In *Proc. EMNLP2016*.

Hoifung Poon and Lucy Vanderwende. 2010. Joint inference for knowledge extraction from biomedical literature. In *Proc. NAACL-HLT2010*.

Sudha Rao, Daniel Marcu, Kevin Knight, and Hal Daumé III. 2017. Biomedical event extraction using abstract meaning representation. In *Proceedings of the BioNLP 2017 workshop*.

Sebastian Riedel, Hong-Woo Chun, Toshihisa Takagi, and Jun'ichi Tsujii. 2009. A markov logic approach to bio-molecular event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*.

Sebastian Riedel and Andrew McCallum. 2011. Robust biomedical event extraction with dual decomposition and minimal domain adaptation. In *Proceedings of the BioNLP Shared Task 2011 Workshop*.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Deepak Venugopal, Chen Chen, Vibhav Gogate, and Vincent Ng. 2014. Relieving the computational bottleneck: Joint inference for event extraction with high-dimensional features. In *Proc. EMNLP2014*.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *Proc. EMNLP2015*.