Learning Latent Structures for Cross Action Phrase Relations in Wet Lab Protocols

Chaitanya Kulkarni, Jany Chan, Eric Fosler-Lussier, Raghu Machiraju
Department of Computer Science and Engineering
Ohio State University
{kulkarni.132,chan.206,fosler-lussier.1,machiraju.1}@osu.edu

Abstract

Wet laboratory protocols (WLPs) are critical for conveying reproducible procedures in biological research. They are composed of instructions written in natural language describing the step-wise processing of materials by specific actions. This process flow description for reagents and materials synthesis in WLPs can be captured by material state transfer graphs (MSTGs), which encode global temporal and causal relationships between actions. Here, we propose methods to automatically generate a MSTG for a given protocol by extracting all action relationships across multiple sentences. We also note that previous corpora and methods focused primarily on local intra-sentence relationships between actions and entities and did not address two critical issues: **(i)** resolution of implicit arguments and **(ii)** establishing long-range dependencies across sentences. We propose a new model that incrementally learns latent structures and is better suited to resolving inter-sentence relations and implicit arguments. This model draws upon a new corpus WLP-MSTG which was created by extending annotations in the WLP corpora for inter-sentence relations and implicit arguments. Our model achieves an F1 score of 54.53% for temporal and causal relations in protocols from our corpus, which is a significant improvement over previous models - DyGIE++:28.17%; spERT:27.81%. We make our annotated WLP-MSTG corpus available to the research community. [1]

1 Introduction

Wet laboratory protocols (WLPs) play an integral role in bioscience and biomedical research by serving as a vehicle to communicate experimental instructions that allow for standardization and replication of experiments. These procedures, typically written in natural language, prescribe actions (Figure 1) to be conducted on materials that generally produce new materials which, in turn, are used by future actions to make newer materials. However, WLPs can be unclear, composed of disconnected and distant parts, and built upon implicit information that were referenced earlier or omitted entirely. Lack of careful documentation has led to a reproducibility crisis (Baker, 2016) in the biosciences and also poses considerable challenges for automation of laboratory procedures: gleaning the effect and semantics of actions requires understanding the underlying experiment, the sentence structure and rationale behind implicitly stated arguments.

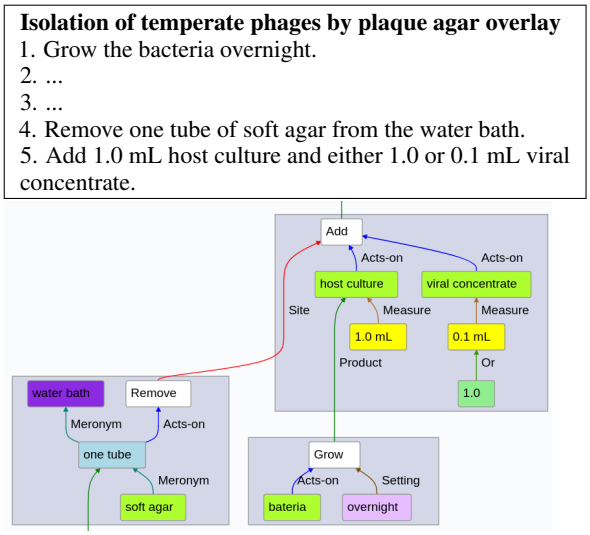


Figure 1: Extraction of a MSTG from an example WLP. The MSTG is composed of Action Graphs (in grey), connected by temporal and causal relationships (e.g., temporal relation "Site" between *Remove* and *Add*). The arrow indicate the direction of material flow.

Currently, there is a dearth of annotated resources for natural language instructions in laboratory protocols. The WLP corpus initially collected by Kulkarni et al. (2018) and later updated by Tabassum et al. (2020) focused solely on relations within sentences. However, actions in WLPs are more complex, containing additional relations between actions (e.g., temporal and causal rela-

[1] The dataset and code is available on the authors' websites

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 6737–6750
August 1–6, 2021.

tions). We propose using material state transfer graphs (MSTG), which are a natural extension of *Action Graphs* (Kulkarni et al., 2018). MSTGs link together several *Action Graphs* into a larger structure by utilizing global temporal and causal relationships that can span several sentences in order to describe the flow of materials from action to action (Section 3). An example of a MSTG is shown in Figure 1. The action phrase *Grow the bacteria overnight* in Step 1 consists of an action *Grow* that *Acts-on* the reagent *bacteria* for an amount of time specified as *overnight*. This *Action Graph* is then connected to other such graphs (like in Step 5) through temporal and causal relationships (e.g., *Grow* action's product is *host culture* thus we use a *Product* link to establish a temporal relation between Step 1 and Step 5).

To automate the generation of MSTGs, we must overcome two distinct challenges prevalent in WLPs. First, the result of a preceding step may not be immediately used by the next step, resulting in long-range dependencies. Second, an action may involve *implicit information*, which is either mentioned earlier or omitted entirely. Current models usually fail to make accurate predictions for long-range relations, as seen in Figure 1 when establishing a temporal relation between Step 1 and Step 5. These methods rely on relation propagation (DyGIE++ Wadden et al. (2019)) or use contextual embeddings (spERT Eberts and Ulges (2019)). Furthermore, neither successfully establish complex relations involving implicit arguments. In Step 5, the *host culture* and *viral concentrate* must be added to the *tube* containing *soft agar* that was removed in Step 4. However, the location *tube* in Step 5 is implicit and has to be correctly inferred to make the *Site* relation between *Remove* and *Add*.

We propose a novel and effective neural network model that: **(i):** uses a series of relational convolutions to learn from relations within and across multiple action phrases and **(ii):** iteratively enriches entity representations with learned latent structures using a multi-head R-GCN model. Our model achieves an F1 score of 54.5% for temporal and causal relations, significantly improving upon previous methods DyGIE++ and spERT for such long-range relations by 26.4% and 26.7% respectively. We analyze our model for intra- and inter-sentence relation extraction and show substantial improvements. Further, we also show the model's ability in resolving implicit arguments to improve temporal relation extraction over the best baseline method by 23.3%.

This paper is organized around two main contributions: **(i):** the WLP-MSTG Corpus that extends the WLP Corpus (Kulkarni et al., 2018) by including intra- and cross-sentence temporal and causal relationships and **(ii):** a novel model that builds upon latent structures to resolve implicit arguments and long-range relations spanning multiple sentences. In Section 2, we describe related works and in Section 3, we introduce MSTGs highlighting the two challenges. Next, we describe our proposed model in Section 4 and demonstrate its performance in Section 5.

2 Related Work

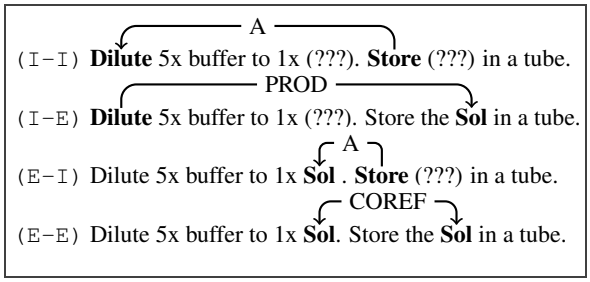
Temporal and Causal Relation Extraction: Prior efforts have shown great promise in learning local and global features (Leeuwenberg and Moens, 2017; Ning et al., 2017). Neural-network-based methods have proven effective (Meng et al., 2017; Meng and Rumshisky, 2018). Notably, Han et al. (2019) use neural support vector machine which can be difficult to train. Early methods for extracting causal relations resorted to feature engineering (Bethard and Martin, 2008; Yang and Mao, 2014). Recently several researchers (Zeng et al., 2014; Nguyen and Grishman, 2015; Santos et al., 2015) used convolutional neural networks (CNNs) for extracting causal features. Notably, Li and Mao (2019) addressed scarcity of training data thorough knowledge-based CNN. However, such methods are not scalable to multiple sentences.

Cross Sentence Relation Extraction: Long range relations are understudied in literature. Prior work focused on relations within a sentence or at best between pairs of sentences (Peng et al., 2017; Lee et al., 2018; Song et al., 2018; Guo et al., 2019). In addition to joint entity and relation extraction models, Wadden et al. (2019) proposed a model that passes useful information across graphs over cross-sentence contexts while Eberts and Ulges (2019) encoded per sentence contextual information for relation extraction over longer sentences.

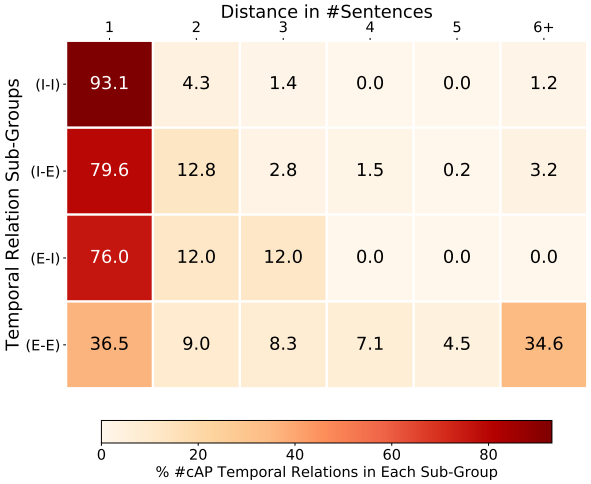
Implicit Arguments: Early methods selected specific features to build linear classifiers (Gerber and Chai, 2010, 2012). Others incorporated additional, manually-constructed resources like named entity taggers and WordNet (Gerber and Chai, 2012; Laparra and Rigau, 2013; Fellbaum,

2012). In contrast, a few notable studies used unlabeled training data to resolve implicit arguments (Chiarcos and Schenk, 2015; Schenk et al., 2016). Finally, Do et al. (2017) explored the full probability space of semantic arguments; however, the method does not scale well.

3 Task Formulation: Material State Transfer Graph



(a) Four different cases of implicit arguments



(b) Distribution of implicit arguments

Figure 2: Implicit Arguments in WLPs. (a) Classification of implicit arguments into four cases. Using the same two actions, we denote the presence of implicit arguments by "(???)" for clarity. (I-I): both product and source are implicit. (I-E): only the product is implied. (E-I): the source is implied. (E-E): both product and source are explicit. (b) Distribution of relations that capture a specific category of implicit arguments. (n = 90 WLPs)

To construct a MSTG from an input protocol, we define the following four concepts. (i) *Action Graphs:* Introduced by Kulkarni et al. (2018), they are extracted from action phrases as seen in Figure 1. Forming the fundamental unit of a MSTG, *Action Graphs* are composed of an *Action*, 17 types of named entities as explicit arguments (e.g, "Reagent", "Location", etc.), and 13 local semantic relations (e.g., "Using", "Measure", "Acts-on",

Data Split	#Docs	#Entities	#iAP	#cAP-TaC
Train	387	34,355	32,585	5,049
Dev	99	13,713	12,578	2,209
Test	128	16,869	15,679	2,724
Total	615	64,937	60,842	9,982

Table 1: Statistics of the Wet Lab Protocol-Material State Transfer Graph Corpus extended with cross Action Phrase Temporal and Causal relationships.

etc.) represented as directed edges, which we shall refer to as *inter-Action Phrase* (iAP) relations hereafter. (ii) *Temporal Relations:* Inspired from prior work (Allen, 1984), we define temporality as a relationship between two action phrases such that an action's product (output) is connected to another action's source (input), thereby imposing a partial or total order. It is also necessary to determine whether an action is executed *before* or *simultaneously* with respect to other actions. We use 5 temporal relations, (namely "Acts-on", "Site", "Coreference", "Product", and "Overlaps") to capture the flow of materials. (iii) *Causal Relations:* Following (Barbey and Wolff, 2007), we define causality as the relationship between two actions where one action directly affects the execution of another action (e.g., if a given action *enables* or *prevents*[2] another action). (iv) *Implicit Arguments:* We characterize implicit arguments into four cases (Figure 2a) depending on whether the source or product of the connected actions is implicit or explicit. Four of the five temporal relations in WLP-MSTG are defined to handle implicit arguments: "Acts-on", "Site", "Coreference", and "Product".

3.1 Corpus for Cross Sentence Relations

Annotation Process: We annotate six-hundred-and-fifteen (615) protocols derived from the WLP Corpus to include the 6 global cross-Action Phrase Temporal and Causal (cAP-TaC) relationships. We split the annotation task into two phases. In the first phase, we worked with 7 expert annotators to develop the guidelines over 8 iterations. Each iteration consisted of 10 protocols that were individually annotated by each expert annotator, and the inter-annotator agreement (IAA) was measured for each of the 10 protocols. At the end of each iter-

[2]Due to the limited instances of "Prevents" relations found in WLPs, we replace these with the relation "Enables". E.g., *Mix regents carefully to not spill contents*, implies a "Prevents" relation from *Mix* to *spill* which is equivalent to an "Enables" relation from *Mix* to *not spill*.

Statistics	WLP-MSTG	W-NUT 2020	WLPC
# Docs	615	615	622
# Entities	64,937	80,659	60,721
# Relations	70,824	54,212	42,425
# Rels/Doc	115.16	88.14	68.20
# cAP-TaC	9,982	-	-

Table 2: Comparison of existing WLP corpora. The WLP-MSTG corpus expands relation coverage by including temporal and causal relationships.

ation, we refined the set of rules to reduce the guidelines' ambiguity. The agreement measured across all annotators using Krippendorff's Alpha (Krippendorff, 2004) on the last iteration was 78.23%.

With a good IAA attained, we began the second phase to collect the train, dev, and test datasets. To ensure the highest quality of the test data, we employed all 7 annotators to work on the same 128 protocols and merged the resulting annotations based on majority voting. In contrast, individual annotators collected the train and dev sets separately to speed up the annotation process. A typical protocol of 30 steps required 25 minutes on average for an annotator to identify all the cAP-TaC relations.

Comparison with previous corpora: Our corpus, WLP-MSTG, extends the WLP corpus (Kulkarni et al., 2018) which was later updated for a WNUT 2020 shared task (Tabassum et al., 2020). WNUT 2020 was primarily designed to facilitate supervised named entity taggers and within-sentence relation extraction methods. We extend the 615 protocols therein to include intra- and inter-sentence temporal and causal relations. To ensure a fully connected graph, we exclude entities and relations annotated for spurious descriptive sentences that do not prescribe any actions (e.g., title, notations, definitions, etc.). Table 2 provides a comparison of statistics among the three corpora.

Analysis: We conducted a distribution analysis of 90 protocols that would typically serve as the *dev* set for machine learning models. Actions connected by temporal and causal relations tend to be consecutive (78.4%); however, a non-trivial number are considerably spaced apart (21.6%) with 1.08% of the total at least 8 actions apart. For implicit arguments, we observed: **(i)** implicit arguments are unusually prevalent in WLPs (88.44%), **(ii)** a higher percentage (55.98%) of the products of an action are implied, and **(iii)** temporally connected actions are closer if they contain implicit arguments; otherwise, they are relatively farther apart Figure 2b. This analysis provides valuable insight about the challenges in the form of long-range relations and implicit arguments that are present in extracting MSTGs from WLPs.

4 A Latent Structure Model for Joint Entity and Relation Extraction

We develop a latent structure model for jointly learning entity and relations within and across multiple sentences. A schematic of the model is shown in Figure 3. In Section 4.1 we describe construction of *span representation* (Figure 3A) from protocol text that incorporates critical features necessary for long-range relation extraction. Section 4.2 explains how the *transcoder block* (Figure 3B) builds upon *latent structures* (as illustrated in Figure 3D) to improve entity and relation representations. Finally, in Section 4.3 we discuss training and regularization strategies to jointly learn span, entity, and relations through a *multi-task loss function* derived from span, entity, and relation scores (Figure 3C). We shall use Figure 1 as a running example throughout the model description.

4.1 Span Representation

Following prior span-based approaches (Wadden et al., 2019; Eberts and Ulges, 2019), our goal is to **(i):** collect a series of tokens from the protocol text, **(ii):** enumerate all spans, and **(iii):** rank top-scoring spans for considerations as candidates for entity and relation extraction.

Token embeddings: We use SciBERT (Beltagy et al., 2019) for learning token representations for a given protocol $\mathcal{P}$. As shown in Figure 3, the input is a protocol $\mathcal{P}$ represented as a collection of sentences $\mathcal{S} = \{s_1, ..., s_{\mathcal{P}}\}$. Each sentence s_i is composed of a sequence of tokens $\{t_1, ..., t_n\}$. For example, within the sentence, *Add 1.0 mL host culture and either 1.0 or 0.1 mL viral concentrate* (Figure 1, Step 5), we identify *host*, *culture*, *and* etc., as the tokens to be passed to the SciBERT model. We batch process sentences in the protocol to generate context-aware embeddings $\{\mathbf{t}_1, ..., \mathbf{t}_n\}$ for each sentence.

Span Enumeration: The spans between two tokens t_i and t_j is represented as $s_{ij} = \{t_i, t_{i+1}, ...t_j\}$. We enumerate all possible spans of upto a size of 10 tokens. For each enumerated span, the span representation $\mathbf{e}_{ij} \in \mathbb{R}^{d_e}$ is derived from

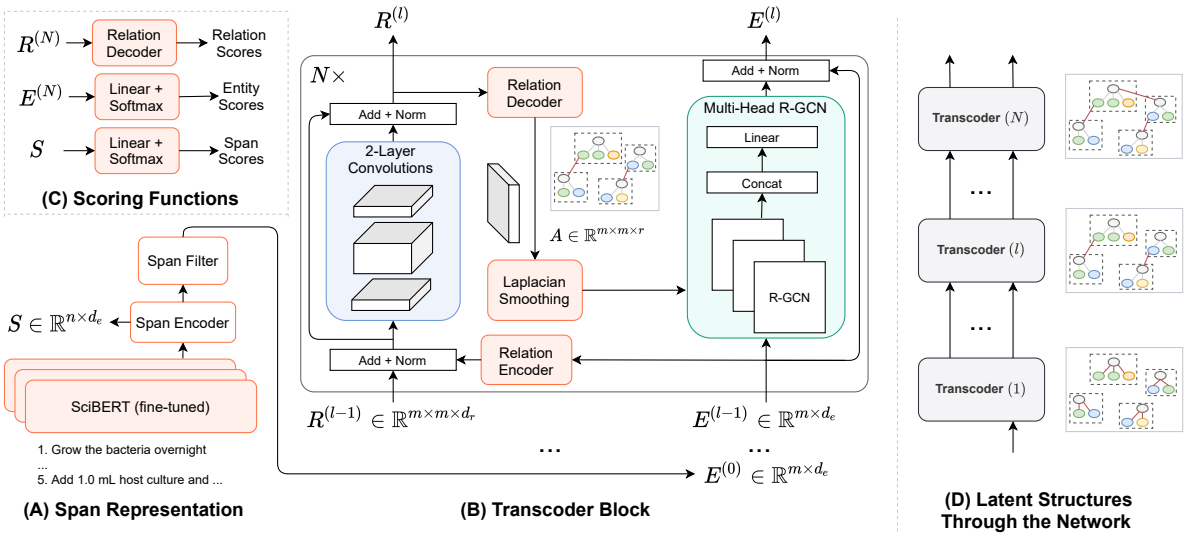


Figure 3: Overview of latent structure model. The model first builds (A) *span representations* (Section 4.1), which are passed into the (B) *transcoder block* (Section 4.2) that leverages (D) *latent structures* to improve entity and relation representations which are scored alongside spans in the (C) *multi-task loss function* (Section 4.3). As indicated in (D), the model first learns simple structures like action graphs. The next set of layers discovers simple temporal and causal relations and uses these connections to discover more complex relations in the final layers.

applying a feed-forward neural network (FFNN) on a concatenation of tokens representations and embeddings:

$$\mathbf{e}_{ij} = \text{FFNN}([\mathbf{t}_i; \mathbf{t}_j; \phi_{sh}(s_{ij}); \phi_{pos}(s_{ij}); \phi_{step}(s_{ij}); \phi_w(s_{ij})]) \quad (1)$$

where, $\mathbf{t}_i$ and $\mathbf{t}_j$ are the first and last token representation. Note, $\phi_{sh}(s_{ij})$ is a soft head representation (Bahdanau et al., 2014) and, $\phi_w(s_{ij})$ is a learnt span width embedding respectively. Further, $\phi_{pos}(s_{ij})$ and $\phi_{step}(s_{ij})$ are two positional embeddings, the former for within sentence while the latter defines the step position within the protocol respectively. Hence, *host culture* and *host culture and* are two valid spans that are enumerated through this process.

Span Pruning: Next, low scoring spans are filtered out during both training and evaluation phases. Following (Lee et al., 2017), the scoring function is implemented as a feed-forward network $\phi_s(\mathbf{e}_{ij}) = \mathbf{w}_s^T \text{FFNN}_s(\mathbf{e}_{ij})$. We rank and pick a number of top scoring spans per sentence by using a combination of **(i):** a maximum fraction $\lambda_p = 0.1$ of spans per sentence, and **(ii):** a minimum score threshold $\lambda_t = 0.5$. Thus, the span *host culture* receives a significantly higher score than *host culture and*, indicating that the former is the correct reagent entity in the prescribed step. These span candidates are then passed to the transcoder block.

4.2 Transcoder Block

In the transcoder block, we propose a novel architecture to improve relation and entity representation from latent structures. The objective is two fold: **(i):** to leverage localized features at phrase and sentence levels to resolve long range relations through a *relation convolutions*, and **(ii):** to learn from latent structures how to resolve implicit arguments through a *multi-head relational graph convolution network* (multi-head R-GCN).

Each transcoder block is composed of a Relation Encoder (Section 4.2.1), Convolution (Section 4.2.2) and Decoder (Section 4.2.3) components, to discover local relationships between the input entities. These relations (represented as latent structures $A \in \mathbb{R}^{m \times m \times r}$) are then passed to the Multi-Head R-GCN (Section 4.2.4) component of the same transcoder block to enrich the entity representation with information about those discovered local relationships. These enriched entities can now be used to predict more complex cross sentence relationships in the next transcoder block. To facilitate deeper networks, we make use of residual connections (He et al., 2016) followed by layer normalization (Ba et al., 2016) as denoted by *Add + Norm* in Figure 3B.

We shall make use of the example (Figure 1), focusing on the long range relationships between Step 1 (i.e., *Grow the bacteria overnight.*) and Step 5 (i.e., *Add 1.0 mL host culture and either 1.0 or*

0.1 mL viral concentrate.) to illustrate the flow of information throughout the transcoder block. The first transcoder block takes as input m high scoring candidate entity span representations (as $E^{(0)} \in \mathbb{R}^{m \times d_e}$) as determined by the pruner [3]. For instance, from Step 1 we identify the following high scoring candidate entities *grow*, *bacteria*, and *overnight* and from Step 5 we find *add*, *1.0 mL*, *host culture*, *0.1 mL*, and *viral concentrate*.

4.2.1 Relation Encoder:

Following (Nguyen and Verspoor, 2019), we make use of a bi-affine pairwise function to encode relations for every pair of entity span representation. That is, we generate relational embeddings for entity pairs like *grow* and *bacteria*, *grow* and *overnight*, etc. Each entity span $\mathbf{e}_{ij} \in \mathbb{R}^{d_e}$ is first projected using two FFNNs to generate the representations $\mathbf{e}^h_{ij} \in \mathbb{R}^{d_h}$ and $\mathbf{e}^t_{ij} \in \mathbb{R}^{d_t}$ indicating the first (head) and the second (tail) argument of a relation:

$$\mathbf{e}^h_{ij} = \text{FFNN}_h(\mathbf{e}_{ij}); \mathbf{e}^t_{ij} = \text{FFNN}_t(\mathbf{e}_{ij})$$

In practice, we batch process all entities to generate $\mathbf{E}_h \in \mathbb{R}^{m \times d_h}$ and $\mathbf{E}_t \in \mathbb{R}^{m \times d_t}$ where m is the number of candidate spans. In our experiments, we let $d_h = d_t$ then use a bi-affine operator to calculate a tensor $\widetilde{\mathbf{R}}^{(l)} \in \mathbb{R}^{m \times d_r \times m}$ for relational embeddings: $\widetilde{\mathbf{R}}^{(l)} = (\mathbf{E}_h \mathbf{L})\mathbf{E}_t^T$. Here $\mathbf{L} \in \mathbb{R}^{d_h \times d_r \times d_t}$ is a learned parameter tensor and d_r is the relation embedding size.

4.2.2 Relation Convolutions:

We enrich the relational embeddings $\widetilde{\mathbf{R}}^{(l)}$ with local relational features within a single phrase (found near the diagonal) and across multiple phrases (found in the upper and lower triangle) using a stack of convolutional layers. We denote $C_w(.)$ to be a $2D$ convolutional operator applying a kernel width of size $w \times w$. In our model, we make use of a two-layer convolution:

$$\mathbf{T}^{(0)} = \text{ReLU}(C_3(\widetilde{\mathbf{R}}^{(l)}))$$
$$\mathbf{R}^{(l)} = \text{ReLU}(C_3(\mathbf{T}^{(0)}))$$

The input $\widetilde{\mathbf{R}}^{(l)}$ is reshaped as $\mathbb{R}^{m \times m \times d_r}$ such that the dimensions d_r acts as the channel dimension in the convolutions. The dimensions of $\mathbf{T}^{(0)}$ is in $\mathbb{R}^{m \times m \times 2d_r}$ with the final output $\mathbf{R}^{(l)} \in \mathbb{R}^{m \times m \times d_r}$.

[3]The entity span representation from the entire subprotocol, (i.e., from steps 1 to 5), are passed as a bag of entities $E^{(0)} \in \mathbb{R}^{m \times d_e}$. However, there aren't any relations (i.e., $R^{(0)}$) to be passed to the first transcoder block

4.2.3 Relation Decoder:

The relational embeddings $\mathbf{R}^{(l)}$ are decoded using a 2-layer FFNN. The decoded scores $\mathbf{A} \in \mathbb{R}^{m \times m \times r}$ captures the latent structures (as shown in Figure 3B). This is re-encoded using the *multi-head R-GCN* to strengthen the model's ability to predict more complex relations in the next transcoder layer.

4.2.4 Multi-head R-GCN:

For each predicted relation score $\mathbf{A}_r \in \mathbb{R}^{m \times m}$, we add self loops and perform Laplacian smoothing (Kipf and Welling, 2017; Li et al., 2018) for normalization following: $\hat{\mathbf{A}}_r = \widetilde{\mathbf{D}}^{-\frac{1}{2}} \widetilde{\mathbf{A}}_r \widetilde{\mathbf{D}}^{-\frac{1}{2}}$ where $\widetilde{\mathbf{A}}_r = \mathbf{A}_r + \mathbf{I}$ and $\widetilde{\mathbf{D}} = \sum_j \widetilde{\mathbf{A}}_{ijr}$. Then, using $\hat{\mathbf{A}}_r$ as an adjacency matrix, we learn multi-head, direction-specific graph convolution transformations. Each head corresponding to a given relation r performs graph convolutions on the entity representation $\mathbf{E}^{(l-1)} \in \mathbb{R}^{m \times d_e}$ to generate $\mathbf{E}^{(l)}_r \in \mathbb{R}^{m \times (d_r/r)}$. A single R-GCN$^{(i)}_r(.)$ (Schlichtkrull et al., 2018) operation for a given relation type r and i^{th} GCN layer corresponds to:

$$\text{R-GCN}^{(i)}_r(\hat{\mathbf{A}}_r, \mathbf{E}^{(i-1)}_r) = \sigma(\hat{\mathbf{A}}_r \mathbf{E}^{(i-1)}_r \mathbf{W}^{(i)}_{fr}) + \sigma(\hat{\mathbf{A}}^T_r \mathbf{E}^{(i-1)}_r \mathbf{W}^{(i)}_{br}) + \mathbf{b}^{(i)}_r \quad (2)$$

where $\mathbf{W}^{(i)}_{fr} \in \mathbb{R}^{d_{i-1} \times d_i}$, $\mathbf{W}^{(i)}_{br} \in \mathbb{R}^{d_{i-1} \times d_i}$ are learnable parameters for incoming and outgoing edge directions respectively and $\mathbf{b}^{(i)}_r$ is the bias. We use the ReLU activation function σ in our networks. As shown in Figure 3B, the outputs of the individual R-GCN heads are concatenated and passed through a FFNN layer to compute the final output $\mathbf{E}^{(l)}$.

For instance, suppose we discovered a local relation in Step 1 between *grow* and *bacteria* after the Relation Decoder component in the first transcoder block. The Multi-head R-GCN takes in the discovered relation (through the latent structure A) and enriches *grow*'s entity embeddings, enabling the next transcoder layer to predict a more complex cross sentence relation between *grow* (Step 1) and *host culture* (Step 5). Since *bacteria* and *host culture* are semantically related, they have similar entity embeddings, and therefore the enriched representation of *grow* (now containing information about *bacteria*) allows for establishing the relation between *grow* and *host culture* in the next transcoder block.

4.3 Training and Regularization

The loss function is a linear combination of cross entropy losses for each of the tasks. We additionally apply label smoothing (Szegedy et al., 2016). The relation extraction is trained on gold entity spans. For regularization, we apply dropout (Srivastava et al., 2014) to the output of each FFNN layer. We make use of dropedge (Rong et al., 2019) for the adjacency matrix $\mathbf{A}_r$ before it is passed to the *multi-head R-GCN* model.

5 Experiments

In contrast to general language models, domain-specific methods have resulted in more competitive baselines and are better suited (Tabassum et al., 2020; Wadden et al., 2019; Eberts and Ulges, 2019) for simultaneously resolving and predicting entities and relations over longer contexts. Thus, we evaluate our model against two state-of-the-art models for jointly predicting entities and relations in scientific-text domain, namely DyGIE++ (Wadden et al., 2019) and spERT (Eberts and Ulges, 2019), on the WLP-MSTG.

We conduct five (5) runs with random initializations for each evaluation and report the test set performance on the model that achieved the median relation F1 score on the dev(elopment) set. All models are evaluated *end-to-end*, where the model takes as input tokenized sentences and predicts all the entities and the relations generating a MSTG. We use the standard precision, recall and F1 metrics. An entity is considered correct if its predicted span and label match the ground truth. Relation extraction is performed on the predicted entity spans. A relation is correct if its relation type and the entity pairs are both correct (in span and type) against the ground truth. We also evaluate our model's performance on WNUT 2020 (Tabassum et al., 2020) corpus. To fairly evaluate relation extraction, we use gold entities to make relation predictions[4] by modifying the loss function to only train on relation scores. We additionally concatenate entity label embeddings to the span representation in Equation (1).

5.1 Results

On the WLP-MSTG corpus, Table 3 shows our best model with $N = 8$ transcoder block layers making modest improvement on entity extraction at 82.0% but improving significantly upon the previous state-of-the-art methods (i.e. DyGIE++ and spERT) in predicting relations. Our model outperforms the baselines for relation extraction with an F1 score on predicting inter-Action Phrase (iAP) relations at 68.0% and cross-Action Phrase Temporal and Causal (cAP-TaC) relations at 54.5%. We further enhanced the performance of our model by sharing the relational decoders' parameters across all layers of the transcoder block (Section 4.2.3). This enables the latent structures to be grounded in output relation types, which also lends itself to be interpretable. The shared relation decoder marginally outperforms the not-shared configuration by 0.5% for iAP relations and 1.1% for cAP-TaC relations.

Short and Long Range Relations: On the WNUT 2020 corpus, which only includes intra-sentence relations, Table 4 shows that our model outperforms the best single model that used the original data by 1.0%. We also report that our model is competitive against the ensemble approach that included models trained on an altered version of the original corpus where they removed duplicate text after clustering. On the WLP-MSTG corpus, we can evaluate both short and long range relations: from Table 3 we see a 3.5% improvement in F1 score over DyGIE++ for iAP relations. This shows that our model leverages the cross-sentence temporal and causal relations that were additionally annotated in WLP-MSTG to improve local iAP relations. Our model outperforms DyGIE++ and spERT on intra-sentence by 4.3% and 26.1% respectively, and significantly improves for inter-sentence cAP-TaC relations by 45.5% and 21.5% respectively. This is attributed to positional embeddings along with the relational convolutions which enables the model to learn intra and inter action phrase relations effectively. We see spERT performing better for "Overlaps" which is largely attributed to the 'CLS' token that spERT embeds to make relation predictions. Figure 4 shows performance on varying the number of sentences in between entities involved in a relation. We observe our model performing the best for all distances between sentences. This is once again attributed to the relational convolution component which is effective in capturing far away relations.

Temporal and Implicit Arguments: In Table 6 we show our model outperforming the baselines for

[4]The best models on WNUT2020 make direct use of gold entities during the training and inference and only focus on relation extraction task.

Models	Action + Entities			iAP Relations (85.2%)			cAP-TaC Relations (14.8%)		
	P	R	F1	P	R	F1	P	R	F1
DyGIE++ (Wadden et al., 2019)	**85.0**	78.5	81.6	66.1	62.9	64.5	**61.5**	18.2	28.1
spERT (Eberts and Ulges, 2019)	76.4	**83.1**	79.6	34.3	59.0	43.4	20.1	45.1	27.8
Our Model (No-Sharing)	82.9	81.2	**82.1**	66.9	68.2	67.5	60.1	48.1	53.4
Our Model (Shared Decoder)	82.8	81.3	82.0	**67.9**	**68.2**	**68.0**	57.8	**51.5**	**54.5**

Table 3: Micro F1 scores for actions + entities and relation extraction (split into iAP and cAP-TaC relations) on the WLP-MSTG test set.

Models	P	R	F1
Miller and Vosoughi (2020)	45.4	86.5	59.6
Single (Sohrab et al., 2020)	80.3	77.4	78.9
Ensemble (Sohrab et al., 2020)	**80.8**	**80.1**	**80.5**
Our Model (single)	80.4	79.3	79.9

Table 4: Micro F1 scores for relation extraction on WNUT 2020 shared task based on gold entities.

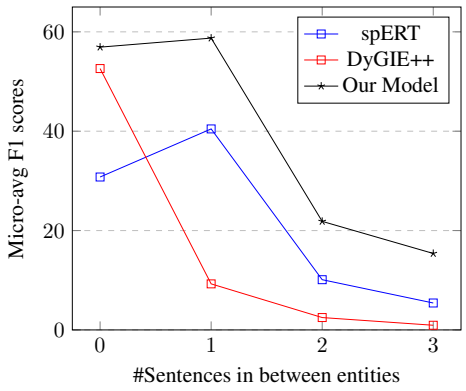


Figure 4: Micro F1 scores for cAP-TaC relation extraction on the test set split by the distance between head and tail entities as measured by number of sentences.

cAP-TaC Relations	DyGIE++	spERT	Ours
Acts-on	62.8	25.4	**66.9**
Site	30.1	22.8	**49.3**
Coreference-Link	6.6	8.8	**23.6**
Product	51.7	45.0	**59.5**
Enables	**62.3**	48.5	61.9
Overlaps	14.7	25.4	**29.1**
Micro F1	52.6	30.8	**56.9**

(a) Intra-Sentence cAP-TaC Relations

cAP-TaC Relations	DyGIE++	spERT	Ours
Acts-on	13.7	35.7	**65.4**
Site	5.4	45.6	**58.2**
Coreference-Link	1.9	4.6	**14.2**
Product	6.8	34.4	**56.2**
Enables	0.0	0.0	0.0
Overlaps	0.0	**2.7**	1.7
Micro F1	7.4	31.4	**52.9**

(b) Inter-Sentence cAP-TaC Relations

Table 5: cAP-TaC relation extraction performance on the test set, split into (a) intra- and (b) inter-sentence relations and presented as per class and micro averaged F1 scores. Bold indicates best performance per row.

temporal relations at 53.4% F1 score. We also observe significant improvements across the board for resolving implicit arguments. We see the highest gains (at 55.6%) compared to the baseline models (1.6% for DyGIE++ and 10.2% for spERT) for (E-I) case (Figure 2a) which only contains 169 samples in the test set. Our model is able to correctly resolve the implicit source (input) to an action by utilizing simple relations that is typically connected to explicit arguments.

Causal relations: The performance for causal relations for our model against DyGIE++ is comparable as seen in Table 6. Causal relations are relatively easier for the baseline models to capture, as they tend to have specific prepositions in between action phrases.[5] However, more complex causal relations are hard. Still, our model is able to deal with such examples, presenting about 0.7% performance gain compared to DyGIE++ and about 10.9% improvement against spERT. This is primarily attributed to the multi-head R-GCN which builds upon simple relations that provide clues to establish harder causal relations. Cross-sentential '*Enables*' relations (as seen in Table 5) are challenging even for our model as once again we do not encode any contextual features.

Model Ablation: Table 7 presents the results of the ablation test of our model on the development set of WLP-MSTG. All three components (i.e., positional embeddings, relation convolutions and

[5]For instance, in Step *Resuspend by vortexing the pellets* baseline models can easily identify an "Enables" relation from *vortexing* to *Resuspend* with the help of the preposition '*by*'.

Groups	DyGIE++	spERT	Ours
Temporal (5034)	23.4	30.1	**53.4**
- (I-I) (1608)	41.5	33.0	**56.4**
- (I-E) (2331)	15.5	35.7	**67.5**
- (E-I) (169)	1.6	10.2	**55.6**
- (E-E) (546)	3.1	5.7	**31.0**
Causal (608)	55.5	45.3	**56.2**

Table 6: Micro F1 scores for cAP-TaC relations split into temporal (and subgroups) and causal relations on WLP-MSTG test set. Refer to Figure 2a for acronym definitions. Bold indicates best performance per row.

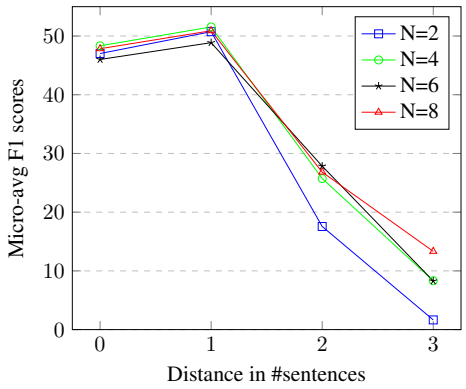


Figure 5: Micro F1 scores for cAP-TaC relation extraction on the dev set for different number (N) of transcoder blocks as illustrated in Figure 3.

multi-head R-GCN) play a significant role in improving cAP-TaC performance. Relation convolutions contributes the most to iAP and cAP-TaC relations by about 1.2% and 2.4% respectively. Positional embeddings impacts iAP relations more (by 1.1%) whereas Multi-Head R-GCN only impacts the more complex relations (cAPTaC by 1.1%) and does not help in improving simpler relations.

How Many Layers?: Figure 5 shows that more layers generally improve far away relations without improving closer ones. This shows that although our model can build upon simple relations that are typically close by, it cannot do the opposite, i.e., leverage far away relations (which are typically more complicated) to improve more challenging closer relations. Our model discovers those complex, distant relations too deep into the network to be utilized to predict the challenging local relations.

Model	All	iAP	cAP
Final Model	**59.5**	**64.3**	**47.5**
- Pos + Step Embedding	58.3	63.2	46.6
- Relation Convolutions	58.0	63.1	45.1
- Multi-head R-GCN	59.2	64.3	46.4
- All above	46.3	52.3	26.0

Table 7: Ablation test of proposed latent structure model evaluated on WLP-MSTG dev set. We present micro F1 scores for both iAP and cAP-TaC relation extraction.

6 Conclusions and Future Work

We present the WLP-MSTG corpus, an extension of the WLP corpus that includes cAP-TaC relationships for building MSTGs. This corpus highlights two unique challenges: (i) the implicit argument problem and (ii) long-range relations. To address these issues, our model builds upon latent structures thus outperforming previous state-of-the-art models for predicting iAP and cAP-TaC relations. We also report significant improvements in understanding implicit arguments and identifying long range relationships across multiple sentences. However, our model's lower absolute performance indicates that we have not fully captured the information needed to facilitate modeling end-to-end workflows, which will have a lasting impact in improving automation in the life sciences and other domains.

References

James F Allen. 1984. Towards a general theory of action and time. *Artificial intelligence*, 23(2):123–154.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Monya Baker. 2016. Reproducibility crisis. *nature*, 533(26):353–66.

Aron K Barbey and Philip Wolff. 2007. Learning causal structure from reasoning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 29.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. 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 3606–3611.

Steven Bethard and James H Martin. 2008. Learning semantic links from a corpus of parallel temporal

and causal relations. In *Proceedings of ACL-08: HLT, Short Papers*, pages 177–180.

Christian Chiarcos and Niko Schenk. 2015. A minimalist approach to shallow discourse parsing and implicit relation recognition. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning-Shared Task*, pages 42–49.

Quynh Ngoc Thi Do, Steven Bethard, and Marie Francine Moens. 2017. Improving implicit semantic role labeling by predicting semantic frame arguments. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 90–99.

Markus Eberts and Adrian Ulges. 2019. Span-based joint entity and relation extraction with transformer pre-training. *24th European Conference on Artificial Intelligence*.

Christiane Fellbaum. 2012. Wordnet. *The encyclopedia of applied linguistics*.

Matthew Gerber and Joyce Chai. 2010. Beyond nombank: A study of implicit arguments for nominal predicates. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1583–1592.

Matthew Gerber and Joyce Y Chai. 2012. Semantic role labeling of implicit arguments for nominal predicates. *Computational Linguistics*, 38(4):755–798.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention guided graph convolutional networks for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 241–251.

Rujun Han, Qiang Ning, and Nanyun Peng. 2019. Joint event and temporal relation extraction with shared representations and structured prediction. 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 434–444.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. *ICLR*.

Klaus Krippendorff. 2004. *Content analysis: An introduction to its methodology*. SAGE Publications.

Chaitanya Kulkarni, Wei Xu, Alan Ritter, and Raghu Machiraju. 2018. An annotated corpus for machine reading of instructions in wet lab protocols. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 97–106.

Egoitz Laparra and German Rigau. 2013. Impar: A deterministic algorithm for implicit semantic role labelling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1180–1189.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692.

Artuur Leeuwenberg and Marie Francine Moens. 2017. Structured learning for temporal relation extraction from clinical records. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1150–1158.

Pengfei Li and Kezhi Mao. 2019. Knowledge-oriented convolutional neural network for causal relation extraction from natural language texts. *Expert Systems with Applications*, 115:512–523.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Yuanliang Meng and Anna Rumshisky. 2018. Context-aware neural model for temporal information extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 527–536.

Yuanliang Meng, Anna Rumshisky, and Alexey Romanov. 2017. Temporal information extraction for question answering using syntactic dependencies in an lstm-based architecture. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 887–896.

Chris Miller and Soroush Vosoughi. 2020. Big green at wnut 2020 shared task-1: Relation extraction as contextualized sequence classification. *arXiv preprint arXiv:2012.04538*.

Dat Quoc Nguyen and Karin Verspoor. 2019. End-to-end neural relation extraction using deep biaffine attention. In *European Conference on Information Retrieval*, pages 729–738. Springer.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48.

Qiang Ning, Zhili Feng, and Dan Roth. 2017. A structured learning approach to temporal relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1027–1037.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph lstms. *Transactions of the Association for Computational Linguistics*, 5:101–115.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2019. Dropedge: Towards deep graph convolutional networks on node classification. In *International Conference on Learning Representations*.

Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. *arXiv preprint arXiv:1504.06580*.

Niko Schenk, Christian Chiarcos, Kathrin Donandt, Samuel Rönnqvist, Evgeny Stepanov, and Giuseppe Riccardi. 2016. Do we really need all those rich linguistic features? a neural network-based approach to implicit sense labeling. In *Proceedings of the CoNLL-16 shared task*, pages 41–49.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *15th International Conference on Extended Semantic Web Conference, ESWC 2018*, pages 593–607. Springer/Verlag.

Mohammad Golam Sohrab, Anh-Khoa Duong Nguyen, Makoto Miwa, and Hiroya Takamura. 2020. mgsohrab at wnut 2020 shared task-1: Neural exhaustive approach for entity and relation recognition over wet lab protocols. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 290–298.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. N-ary relation extraction using graph-state lstm. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2226–2235.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Jeniya Tabassum, Wei Xu, and Alan Ritter. 2020. Wnut-2020 task 1 overview: Extracting entities and relations from wet lab protocols. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 260–267.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. 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 5788–5793.

Xuefeng Yang and Kezhi Mao. 2014. Multi level causal relation identification using extended features. *Expert Systems with Applications*, 41(16):7171–7181.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344.

7 Appendix

7.1 Material State Transfer Graph Example

We describe a full material state transfer graph (as seen in Figure 7) designed for the protocol in Figure 6. Each action phrase found in the protocol text is converted into an action graph (as seen in grey boxes in Figure 7). For example, the action phrase: *Grow the bacteria overnight*, we identify an "Action" *Grow* and all of its arguments like "Reagent" *bacteria* and "Time" *overnight*. These actions and entities are interconnected with local relations that we call iAP (inter-Action Phrase) relations. For instance, relations like "Acts-on" between *Grow* to *bacteria* and "Setting" from *Grow* to *overnight* to indicate that is how long we should be growing the bacteria. Then, the action phrases as action graphs are interconnected with cross-Action Phrase Temporal and Causal (cAP-TaC) relations. These relations can connect to any action or entity in the action graph as seen in Figure 7. For example, the "Product" relation from *Grow* to *host culture* indicates two things, (i) that the actual product of the *Grow* "Action" is *host culture* and (ii) the steps involving *Grow* must take place first before the "Action" *Add*. We carefully define each relation used as cAP-TaC relations below:

7.2 Temporal Relations

The following four relations behave as "before" temporal relations (ie "Acts-on", "Site", "Product", "Coreference"). Whereas the fifth relation "Overlaps" is used when any two actions have any degree of overlap in time. The four relations also define which implicit argument relation group do they fall under. The four groups are (i)(I-I) as in both the product and the sources are implicit. (ii)(E-I) The product is explicit, but the source is implied. (iii) (I-E) the product is implied but the source is explicit. And, (iv) (E-E) both the source and product are explicit.

Acts-on: Connects to a previous "Action" if the product of that "Action" is implicit. Otherwise directly connects to the named entity (which can be a "Reagent", "Location", "Device" etc) that the previous Action has a "Product" relation to. If directly connected to an "Action", this would fall under (I-I) case of implicit argument types. If connected to any named entity which is a product of the previous action then it would fall under (E-I) case.

Isolation of temperate phages by plaque agar overlay

1. Grow the bacteria overnight.
2. Melt soft agar overlay tubes in boiling water.
3. Place in the 47C water bath.
4. Remove one tube of soft agar from the water bath.
5. Add 1.0 mL host culture and either 1.0 or 0.1 mL viral concentrate to the tube.
6. Mix the culture contents in the tube well by rolling back and forth between two hands.
7. Immediately empty the tube contents onto an agar plate.
8. Sit RT for 5 min.
9. Gently spread the top agar over the agar surface by sliding the plate on the bench surface using a circular motion.
10. Harden the top agar by not disturbing the plates for 30 min.
11. Incubate the plates (top agar side down) overnight to 48 h.
12. Temperate phage plaques will appear as turbid or cloudy plaques, whereas purely lytic phage will appear as sharply defined, clear plaques.

Figure 6: An example experimental protocol. The first 11 steps contain imperative phrases, while the last sentence describes the end results and their subsequent utilization. A full material state transfer graph for this protocol is shown in Figure 7

Site: Similar to "Acts-on", this relation links to the previous "Action" if the product of that "Action" is implicit, and that product is where the current "Action" is taking place. Otherwise we directly connect to the appropriate named entity. Once again, similar to "Acts-on", if directly connected to "Action", it falls under (I-I) case, otherwise its (E-I) case.

Product: This relation is used to identify the product of the current "Action", either its found in its own action phrase or in some future action phrase. If the product is identified within its action phrase (which is quite rare) it would be considered an iAP relation. Otherwise, this would fall under (I-E) case.

Coreference: This is used when the objects or arguments are in the same state. We connect the object to the same object referred before only if that object has not undergone any transformations by any actions in between. This relation falls under (E-E) case.

Overlaps: This relation is used to indicate which two actions are being performed simultaneously or that have any degree of overlap between them in terms of time.

7.3 Causal Relation

We only make use of one causal relation type "Enables". Due to low numbers on "Prevents" relations, we turn them into an "Enables" relation by

simply negating the "Action" involved in the relationship. For example, *Mix regents carefully to not spill contents*, we replace a "Prevents" relation from *Mix* to *spill* with an "Enables" relation from *Mix* to *not spill*. In many elaborate negative words we make use of "Mod-Link" to connect to the additional descriptors to the relevant action.

7.4 Implementation Details

In evaluating on WLP-MSTG, we overcome memory limitations in baseline models during training and inferencing, we sub-divide long protocols into overlapping windows of 5 sentences each, with a stride of 2 (i.e., each consecutive window shares 3 sentences). To ensure fair comparison we also incorporate this restriction to our model, although our models is capable of a much larger window size. The final evaluation is done by merging the predictions in the form of sub-graphs into one complete material state transfer graph (MSTG) and resolving duplicate predictions through majority voting. We identify duplicates through exact match of spans boundaries for entities and exact match of entity span and its types for relations.

Hyperparameters We make use of Adam optimizer with a initial learning rate of 2.13×10^{-5}. For generating span candidates we only enumerate them upto 10 tokens in width. We set the positional embedding $\phi_{pos}(s_{ij})$ table size to 100. For step embedding $\phi_{step}(s_{ij})$ we only learn embeddings for 5 steps. Both embeddings use embedding dimensions as 50. The span embedding size $d_e = 340$, and the relational embedding size d_r is set to 100. Label smoothing [symbol] is set to the default value of 0.1. Dropout used in every FFNN has $p = 0.2$ and the dropedge used right before multi-head R-GCN model is set with $p = 0.5$.

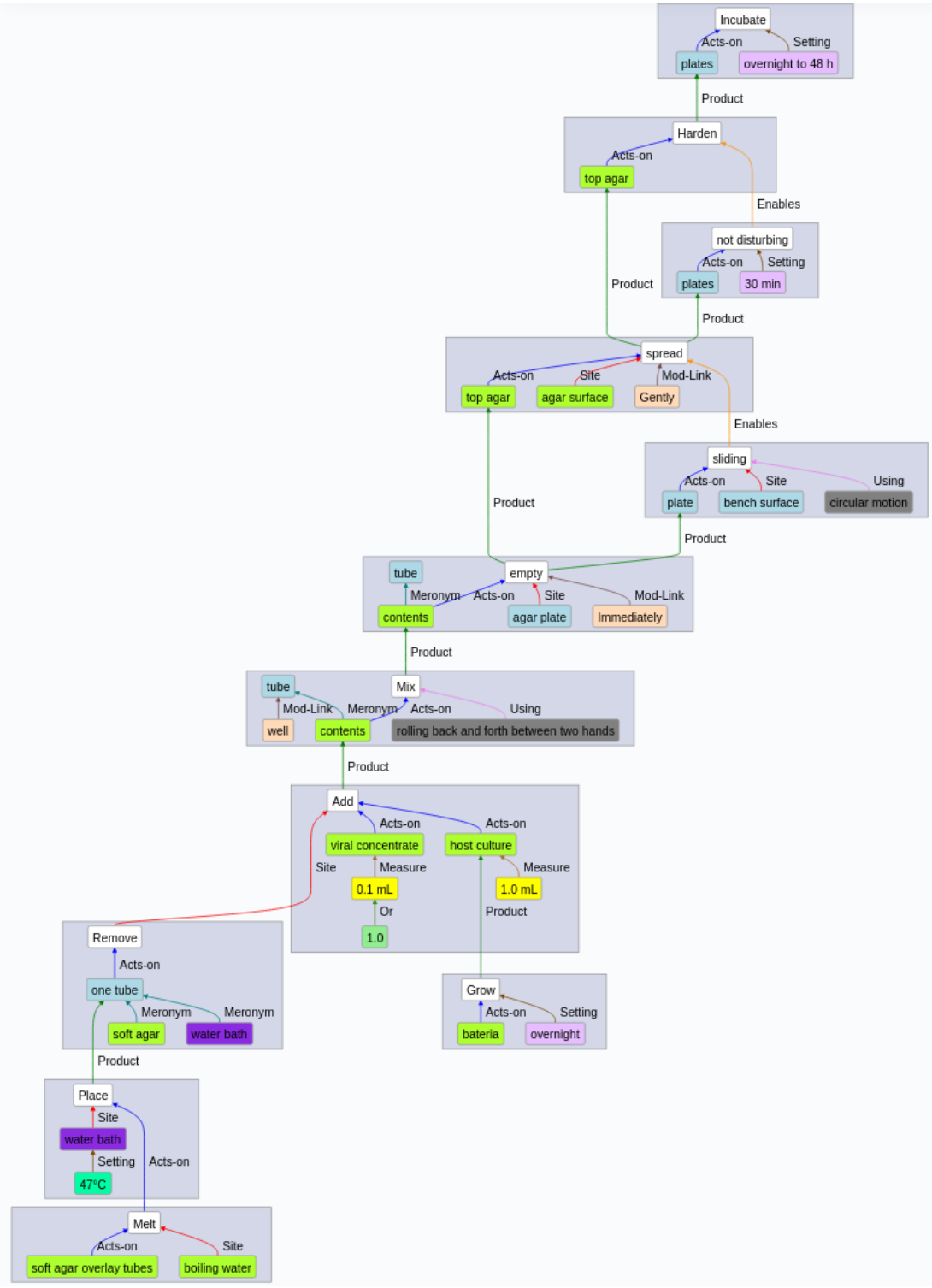


Figure 7: A full material state transfer graphical representation of the example protocol in Figure 6.