Biomedical Event Extraction using Abstract Meaning Representation

Sudha Rao[1], **Daniel Marcu**[2], **Kevin Knight**[2], **Hal Daumé III**[1]
[1]Computer Science, University of Maryland, College Park,
[2]Information Sciences Institute, University of Southern California
raosudha@cs.umd.edu, marcu@isi.edu, knight@isi.edu, hal@cs.umd.edu

Abstract

We propose a novel, Abstract Meaning Representation (AMR) based approach to identifying molecular events/interactions in biomedical text. Our key contributions are: (1) an empirical validation of our hypothesis that an event is a subgraph of the AMR graph, (2) a neural network-based model that identifies such an event subgraph given an AMR, and (3) a distant supervision based approach to gather additional training data. We evaluate our approach on the 2013 Genia Event Extraction dataset[1] (Kim et al., 2013) and show promising results.

1 Introduction

For several years now, the biomedical community has been working towards the goal of creating a curated knowledge base of biomolecule entity interactions. The scientific literature in the biomedical domain runs to millions of articles and is an excellent source of such information. However, automatically extracting information from text is a challenge because natural language allows us to express the same information in several different ways. The series of Genia Event Extraction shared tasks (Kim et al., 2009, 2011, 2013, 2016) has resulted in various significant approaches to biomolecule event extraction spanning methods that use learnt patterns from annotated text (Bui et al., 2013) to machine learning methods (Björne and Salakoski, 2013) that use syntactic parses as features. In this work, we find that a semantic analysis of text that relies on Abstract Meaning Representations (Banarescu et al., 2013) is highly useful because it normalizes many lexical and syntactic variations in text.

[1]This dataset is different from BioNLP 2016 GE dataset

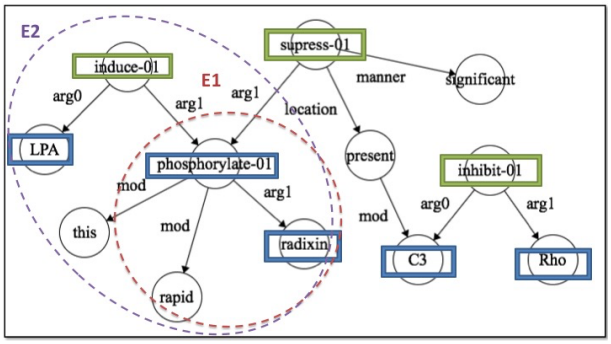

Figure 1: AMR with sample event annotations for sentence *"This LPA-induced rapid phosphorylation of radixin was significantly suppressed in the presence of C3 toxin, a potent inhibitor of Rho"*

AMR is a rooted, directed acyclic graph (DAG) that captures the notion of *who did what to whom* in text, in a way that sentences that have the same basic meaning often have the same AMR. The nodes in the graph (also called concepts) map to words in the sentence and the edges map to relations between the words. In the recent past, there have been several efforts towards parsing a sentence into its AMR (Flanigan et al., 2014; Wang et al., 2015; Pust et al., 2015; May, 2016). AMR naturally captures hierarchical relations between entities in text making it favorable for complex event detection. For example, consider the following sentence from the biomedical literature: *"This LPA-induced rapid phosphorylation of radixin was significantly suppressed in the presence of C3 toxin, a potent inhibitor of Rho"*. Figure 1 shows its Abstract Meaning Representation (AMR). The subgraph rooted at phosphorylate-01 identifies the event E_1 and the subgraph rooted at induce-01 identifies the event E_2 where

$E_1 =$ *phosphorylation of radixin*;
$E_2 =$ *LPA induces* $E1$.

We hypothesize that an event structure is a sub-

Proceedings of the BioNLP 2017 workshop, pages 126–135,
Vancouver, Canada, August 4, 2017.

Type	Primary Args.
Gene_expression	T(P)
Transcription	T(P)
Localization	T(P)
Protein_catabolism	T(P)
Binding	T(P)+
Phosphorylation	T(P/Ev), C(P/Ev)
Regulation	T(P/Ev), C(P/Ev)
Positive_regulation	T(P/Ev), C(P/Ev)
Negative_regulation	T(P/Ev), C(P/Ev)

Table 1: Event types and their arguments in the 2013 Genia Event Extraction task

graph of a DAG structure like AMR and under this assumption, we cast the event extraction task as a graph identification problem. Our **first contribution** is the testing of the above hypothesis that an event structure is a subgraph of an AMR graph. Given a sentence, we automatically obtain its AMR using an AMR parser (Pust et al., 2015) and explain how an event can be defined as a subgraph of the AMR graph. Under the assumption that we can correctly identify such an event subgraph from an AMR graph when it exists, we evaluate how good is our definition (Section 2).

Our **second contribution** is a supervised neural network-based model that is trained to identify an event subgraph given an AMR (Section 3). Our model is built on the intuition that the path between an interaction term and an entity term in an AMR graph contains important signal for identifying the relation between them. For e.g. in figure 1 the path {*'induce-01', 'arg0', 'LPA'*} suggests that *LPA* is the cause of *induce*. We encode this path using word embeddings pre-trained on millions of biomedical text and develop two pipelined neural network models: (a) to identify the *theme* of an *interaction*; and (b) to identify the *cause* of the *interaction*, if there exists one.

Experimental results show that our model, although achieves a reasonable precision, suffers from low recall. Our **third contribution** is a distant supervision (Mintz et al., 2009) based approach to collect additional annotated training data. Distant supervision works on the assumption that given a known relation between two entities, a sentence containing the two entities is likely to express this relation and hence can serve as training data for that relation. Data gathered using such a method can be noisy (Takamatsu et al., 2012). Roth et al. (2013) have discussed several prior work that address this issue. In our work, we introduce a method based on AMR path heuristic

This LPA-induced rapid phosphorylation of radixin was significantly suppressed in the presence of C3 toxin, a potent inhibitor of Rho

T1	(Protein, LPA)
T2	(Protein, radixin)
T2	(Protein, C3)
T4	(Protein, Rho)
T5	(Phosphorylation, phosphorylate)
T6	(Positive_regulation, induce)
T7	(Negative_regulation, suppress)
T8	(Negative_regulation, inhibit)
E1	(Type: T5, Theme: T2)
E2	(Type: T6, Theme: E1, Cause: T1)
E3	(Type: T7, Theme: E1)
E4	(Type: T8, Theme: T4, Cause: T3)

Table 2: Example event annotation. The protein annotations T1- T4 are given as starting points. The task is to identify the events E1-E4 with their interaction type and arguments.

to selectively sample the sentences we obtain using distant supervision (Section 3) and show its effectiveness over our vanilla neural network model.

We evaluate our event extraction model on the 2013 Genia Event Extraction dataset and show that our model achieves promising results when compared to the state-of-the-art system. Given that AMR parsing is still a young field, our model, which currently uses a parser of 67% accuracy, would perform better with improved AMR parsers.

2 AMR based event extraction model

2.1 Task description

The biomedical event extraction task in this work is adopted from the Genia Event Extraction subtask of the well-known BioNLP shared task ((Kim et al., 2009), (Kim et al., 2011), (Kim et al., 2013)). Table 2 shows a sample event annotation for the sentence in Figure 1. The protein annotations T1- T4 are given as starting points. The task is to identify the events E1-E4 with their interaction type and arguments. Table 1 describes the various event types and the arguments they accept. The first four event types require only unary theme argument. The binding event can take a variable number of theme arguments. The last four events take a theme argument and, when expressed, also a cause argument. Their theme or cause may in turn be another event, creating a nested event (For e.g. event E2 in Table 2).

2.2 Model description

We cast this event extraction problem as a subgraph identification problem. Given a sentence we

first obtain its AMR graph automatically using an AMR parser (Pust et al., 2015). Next, we identify protein nodes and interaction nodes in the graph.

Protein Node Identification: In both the training and the test set, protein terms are pre-annotated (e.g. $T1$ to $T4$ in Table 2). We then use the AMR graph alignment information to identify nodes in the AMR graph aligned to these protein terms to get our protein nodes P.

Interaction Node Identification: In the training data, interaction terms are pre-annotated (e.g. $T5$ to $T8$ in Table 2). To identify the interaction terms in the test set we use the following heuristic: any term that was annotated as an interaction term more than once in the training data is considered as an interaction term in the test data as well. We then use the AMR graph alignment information to identify nodes in the AMR graph aligned to the interaction terms to get our interaction nodes T.

Given P and T, we identify an event sub-graph using the following two-step process:

a. Theme Identification: Every pair (p_i, t_j) where $p_i \in P$ and $t_j \in T$, is a candidate for an event e_m defined as e_m*: (Type:* t_j*, Theme:* p_i*)* where *Type* is one of the nine event types in Table 1. If e_m can take other events as arguments (last four event types in Table 1) and if the shortest path between t_j and p_i includes an interaction term t_k, such that the pair (p_i, t_k) is an event e_n in itself, then we define the event e_m instead as e_m*: (Type:* t_j*, Theme:* e_n*)*. For e.g. in Figure 1, the path between *induce-01* and *radixin* includes *phosphorylate-01* which is an event in itself (E_1). Hence event E_2 is defined with E_1 as its theme (in Table 2).

b. Cause Identification: For events e_m*: (Type:* t_j*: Theme:* p_i*)* that can take a cause argument, we identify possible candidates for their cause by again looking for all pairs (p_l, t_j) where $p_l \in P$ and $l \neq i$ and add cause to the event e_m as e_m*: (Type:* t_j*, Theme:* p_i*, Cause:* p_l*)*. Since these events can even take other events as their cause argument, we identify additional candidates for their cause by looking for all pairs (e_n, t_j) where $e_n \in E$ and $n \neq m$ and add cause to the event e_m as e_m*: (Type:* t_j*, Theme:* p_i*, Cause:* e_n*)*.

2.3 Upper bound using *"event is a subgraph of AMR"* hypothesis

Before we learn to identify event sub-graphs from an AMR graph, we first calculate the upper bound that we are setting for our model because we are using an AMR parser instead of obtaining gold AMRs. For calculating this upper bound, we first obtain the AMR graph of a sentence using the AMR parser and then assume that if an event is a sub-graph of this AMR graph then we can identify it correctly. Table 3 shows the upper bound we get on the dev set of the 2013 Genia Event Extraction dataset (described in Section 5.1). The last column in the table is the state-of-the-art F1 score obtained by the system EVEX (Hakala et al., 2013) on the test set of the dataset[2].

Event Type	R	P	F1	F1 ()
Gene_expression	87.82	100.00	93.51	
Transcription	65.31	100.00	79.01	
Localization	86.80	100.00	92.93	
Protein_catabolism	90.00	100.00	94.74	
==[SVT-TOTAL]==	82.48	100.00	90.04	76.59
Binding	67.83	95.83	79.43	42.88
Phosphorylation	60.62	80.14	69.03	65.37
Regulation	42.61	61.73	50.42	
Positive_regulation	41.93	65.43	51.11	
Negative_regulation	50.94	65.85	57.45	
==[REG-TOTAL]==	45.16	64.33	53.00	38.41
==[ALL-TOTAL]==	65.98	85.44	74.18	50.97

Table 3: Upper bound on the dev set using our *"event is a subgraph of AMR"* hypothesis

In case of simple events i.e. events that take only proteins as theme arguments, an event is always a subgraph of the AMR unless there is an alignment error causing the protein node or the interaction node to be missing. Hence the upper bound on our precision is 100% whereas the upper bound on our recall is 82.48% for these simple events. In case of the other event types where an event can take other events as arguments, an event is correctly identified only if the path between the pair (p_i, t_j) in the AMR graph includes all its sub-events. Therefore we lose more on the precision and recall in these cases due to AMR parsing errors bringing our overall upper bound on precision down to 85.44% and our overall upper bound on recall down to 65.98%. These results give us following two important insights:

1. By using this hypothesis we have set an upper bound of 74.18% F1-score for our learning model.
2. As the accuracy of automatic AMR parsers improve, our model will perform better at the event extraction task.

[2]We compare our numbers on the dev set to the EVEX numbers on test set since gold annotations for the test set are not available for download

3 LSTM based learning model

In this section we will describe our model that learns to identify an event sub-graph from an AMR graph. The key idea is that the path between the interaction node and the entity node (where the term entity is used to denote both a protein and a sub-event) contains information about how the event is structured. We build on this idea to develop a supervised model using Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) architecture that can learn to identify events using the nodes and the edges in the AMR path between the interaction term and the entity term.

3.1 Motivation

The input to our problem is a sequence of words (w_i) interwound with edge labels (e_j) of the form: $w_1, e_1, w_2, e_2, ..., e_{n-1}, w_n$ that exists in the path between an interaction node and an entity node in an AMR graph. Due to large semantic variations that exist in naturally occurring texts, traditional feature based methods suffer from sparsity issues while learning from such a sequence. Neural network based models provide a framework for learning from non-sparse representations. Specifically, LSTM is known to handle sequences of variable length and capture long range dependencies well. Since the input sequence in our case falls into this category, we build our model using the LSTM framework.

3.2 Event identification

We model the event identification task as a two-step process: *Theme Identification* and *Cause Identification*. For simple events, this process includes only theme identification (since they don't have cause). We describe the two LSTM models corresponding to the two steps as follows:

3.2.1 Theme Identification

Given a pair of interaction node (t_j) and protein node (p_i), the task is to identify if there exists an event with t_j as the interaction and p_i as the theme; and if yes, what is the type of the event. We cast this problem as a multi-class classification task with label set as $L : \{NULL \cup Event\_types\}$ where *Event_types* correspond to the nine event types described in Table 1 and *NULL* corresponds to no event. We train an LSTM model for this task with the input layer as the embeddings corresponding to the sequence of words interwound with edge labels in the shortest path between p_i and t_j in the AMR graph. We use a hidden layer of size 100 and an output layer of the size of our label set L. For e.g. in Figure 2, the sequence {*'phosphorylate-01', 'arg1', 'radixin'*} is the input sequence and the event type *Phosphorylation* is its label.

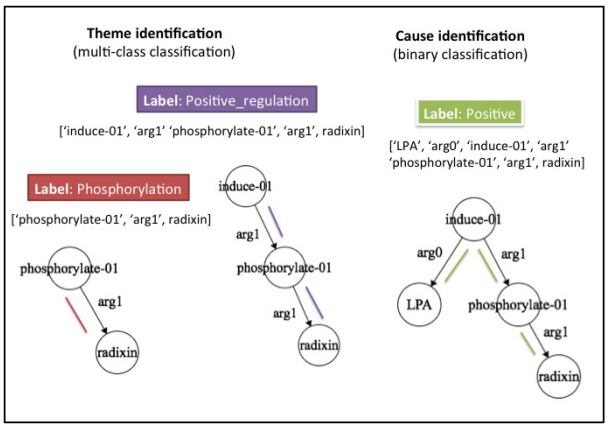

Figure 2: Theme identification and Cause identification stages

3.2.2 Cause Identification

The last four event types in Table 1 can take proteins or other events as cause argument. We cast this problem as a binary classification task where for an event we ask the question if a protein/event is its cause argument or not for every protein and every other event in that sentence. Let e_m be the event identified as $e_m : (Type : t_j, Theme : p_i)$ that can take a cause argument. Let $C = P \cup E$ where P is the set of all other proteins in the AMR graph (except p_i) and E is the set of all identified events (except e_m). For every $c_k \in C$, we get the shortest path between c_k and t_j and combine it with the shortest path between p_i and t_j and use the words and edges in this combined path as the input layer of our second LSTM model. We use a hidden layer of size 100 and an output layer of size one corresponding to the binary prediction of whether c_k is the cause of the event e_m or not.

3.3 Initialization of Embeddings

When initializing our model, we have two choices: we can initialize the embeddings in the input layer randomly or we can initialize them with values that reflect the meanings of the word types. It has been seen that using pre-defined word embeddings improves the performance of RNN models over random initializations (Collobert and Weston,

2008; Socher et al., 2011). We initialize the vectors corresponding to words in our input layer with 100-dimensional vectors generated by a word2vec (Mikolov et al., 2013) model trained on over one million words from the PubMed central article repository. Words not included in the pre-trained model and the edges are initialized randomly using uniform sampling from [-0.25, +0.25] to match the embedding standard deviation.

3.4 Event Construction

During test time, we first make predictions using our LSTM model for Theme identification. For every pair (p_i, t_j) with a non-zero label l, we construct events as follows: For label l corresponding to interaction types that take only proteins as theme arguments, we construct event as $e_m : (Type : t_j, Theme : p_i)$. For label l corresponding to interaction types that can take another event as its theme, we look at the path between t_j and p_i in the AMR. If this path includes a pair (t_k, p_i) that has a non-zero label, then we construct an event $e_n : (Type : t_j, Theme : e_p)$ where e_p is the event constructed from the pair (t_k, p_i). Otherwise, we construct the event as $e_n : (Type : t_j, Theme : p_i)$.

For each of the predicted event $e_m : (Type : t_j : Theme : p_i)$ that can take a cause argument, we run the second LSTM model for its Cause identification. If there is a pair (p_i, c_k) which has a positive label, then we assign c_k as the cause of the event e_m.

4 Distant Supervision

An empirical evaluation of our LSTM-based learning model (Section 5.4) shows that it can suffer from low recall. Obtaining additional human annotated data for our complex event extraction task can be very costly. This motivates us to develop an approach that can gather more training data with minimal supervision.

4.1 Motivation

Distant supervision as a learning paradigm was introduced by Mintz et al. (2009) for relation extraction in general domain. They use Freebase to get a set of relation instances and entity pairs participating in those relations, extract all sentences containing those two entity pairs from Wikipedia text and use these sentences as their training data. This work and many others show that distant supervision technique yields significant improvements in relation extraction. Neural network models like LSTM need to be trained on substantial amounts of training data for them to be able to generalize well. However due to lack of labeled data in biomedical domain, most work in relation extraction in this domain has been restricted to purely supervised techniques. In this work we cope with this problem by gathering additional training data using distant supervision from a knowledge base.

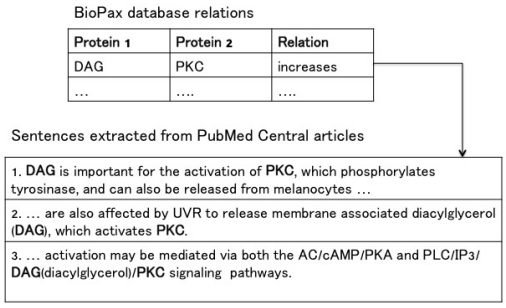

Figure 3: Distant Supervision: Sentences extracted from PubMed Central articles using BioPax database relations

4.2 Methodology

Relation extraction using distant supervision requires two things: 1) A knowledge base containing relations between proteins, and 2) A large corpus of unannotated text that contain protein mentions. We use the BioPax (Biological Pathway Exchange) database (Demir et al., 2010) as our knowledge base of protein relations and we use the PubMed central articles as our unannotated text corpus. Given a database entry of the form *('Protein1', 'Protein2', 'relation')*, we extract all sentences from the PubMed central articles in which the two proteins co-occur. For example, Figure 3 shows some sample sentences extracted for the database entry *('DAG', 'PKC', increases)*. The first two sentences in the figure indeed express the relation in the database but the third sentence just mentions the two proteins in a comma separated list. We observe that a lot of the extracted sentences fall into the category of the third sentence. Hence as a first step, we filter such instances by tagging the sentence with their parts-of-speech and removing those in which the two proteins are separated only by nouns (or punctuations).

4.3 AMR Path Based Selection

The traditional distant supervision approach says that all the sentences extracted using the method above can be used as additional training data un-

Event Type	Biopax relation
Gene_expression	adds_modification
Transcription	adds_modification
Localization	adds_modification
Protein_catabolism	adds_modification
Binding	binds
Phosphorylation	adds_modification
Regulation	increases, increases_activity
Positive_regulation	increases, increases_activity
Negative_regulation	-

Table 4: Mapping between event types and Biopax model relations

der the assumption that all sentences in which the proteins co-occur express the relation mentioned in the database. However Takamatsu et al. (2012) note that this approach can often lead to a lot of false positives. Roth et al. (2013) have discussed several prior work that try to reduce such noise in the data. In our work, we develop a novel selection technique for reducing such noise using AMR path heuristic. We make the observation that given two protein nodes in an AMR, if there is a relation r between the two then the shortest path between the two protein nodes in the AMR contains the interaction term expressing the relation r.

For e.g. Figure 4 shows the AMR for the sentence *"DAG is important for the activation of PKC, which phosphorylates tyrosinase, and can also be released..."* that was extracted using the database entry {*'DAG', 'PKC', 'increases'*}. The interaction term *'activate'* suggesting the relation 'increases' exists in the shortest path between the proteins *DAG* and *PKC*. Figure 5 shows AMR for the sentence *"The sun-network links TCF3 with ZYX and HOXA9 via NEDD9 and CREBBP, respectively."* extracted for the pair *('TCF3', 'HOXA9', increases)*. There is no interaction term suggesting the relation 'increases' in the shortest path between the proteins *TCF3* and *HOXA9*.

Table 4 shows the mapping we define between the event types and the relations found in the entries *('Protein1', 'Protein2', 'relation')* that we extracted from the Biopax model. In each sentence extracted for the database entry *('P_1', 'P_2', 'r')* , we check if the shortest path between the two protein nodes P_1 and P_2 in the AMR of the sentence contains one of the interaction terms corresponding to the event type mapped to the relation r. We discard all those sentences that do not satisfy this constraint.

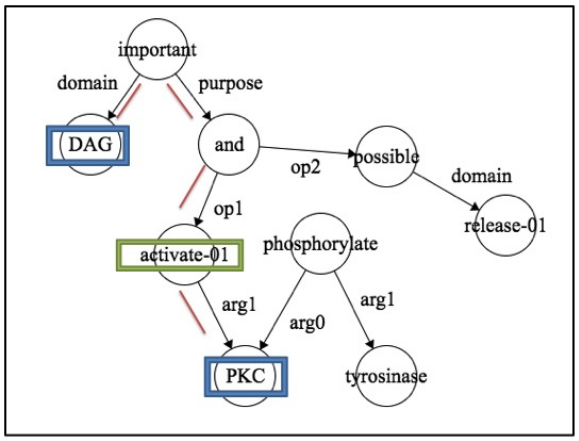

Figure 4: Interaction term *'activate'* corresponding to the relation 'increases' exists in the shortest path between *DAG* and *PKC*

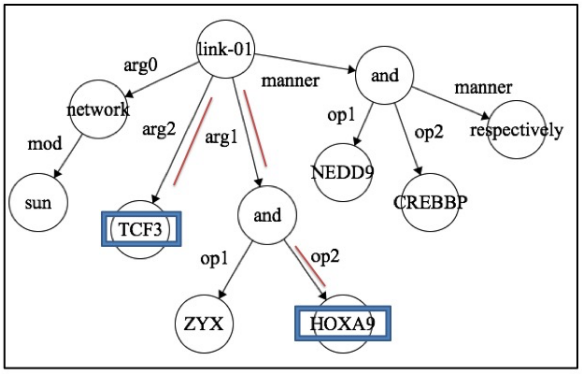

Figure 5: No interaction term corresponding to the relation 'increases' exists in the shortest path between *TCF3* and *HOXA9*

4.4 Using Data for LSTM Model

We use these selected sentences as additional training data for our two LSTM models as follows:
a. Theme identification: Let S be the sentence extracted for the database entry *('DAG', 'PKC', 'increases')* and let '$activates$' be the interaction term that exists in the shortest path between the protein nodes. Since the database entry refers to *'DAG'* as the cause and *'PKC'* as the theme, we assume these roles for the two proteins in the extracted sentence S as well. Therefore, we can now use the path between the interaction term *'activates'* and the theme *'PKC'* as an input sequence for our model with the label corresponding to the event type of the interaction term *'activates'*.
b. Cause identification: In case of cause identification instead of using the path between the interaction term and the theme entity, we use the shortest path between the cause entity and the theme entity via the interaction term and use this as an input sequence to our model with a positive label.

5 Experiments

5.1 Dataset and task setting

The event extraction task described in this work corresponds to the Task 1 of the Genia Event Extraction task described by the BioNLP Shared Task series (2009, 2011 and 2013). We train a model on a combination of *abstract collection* (from 2009 edition) and *full text collection* (from 2011 and 2013). We test our model on the dev set of the 2013 edition (since the gold annotation is publicly available only for the dev set and not the test set).

5.2 Data prepraration

The dataset made available for the Shared Task is in the form of sentences and event annotations as shown in Table 2. We convert these event annotations into input sequences and labels for our multi-class classification task (theme identification) and for our binary classification task (cause identification) as follows

a. Theme identification: Given a sentence, we define the set T as the set of interaction terms corresponding to all its event annotations. We define the set P as the set of all its protein mentions. For every pair (t_j, p_i) where $p_i \in P$ and $t_j \in T$, we create a training data of the form $\{w_1, e_1, w_2, e_2, ..., e_{n-1}, w_n, label\}$ where the input sequence corresponds to the words interwound with edge labels in shortest path between t_j and p_i; and the $label$ is the event type of the event e_m if there exists an event $e_m : (Type : t_j, Theme : p_i)$, *NULL* otherwise. We create the test data similarly; except we do not use event annotations for creating the set T but instead identify terms in the sentence that was annotated as an interaction term in the training data more than once.

b. Cause identification: For every pair (t_j, p_k) where t_j is part of some event annotation $e_m : (Type : t_j, Theme : p_i)$ of event type that can take cause argument and $p_k \in P$, we create a training data of the form $\{w_1, e_1, w_2, e_2, ..., e_{n-1}, w_n, label\}$ where the input sequence corresponds to the shortest path between p_k and p_i via t_j; and the label is 1 if p_k is the cause of the event e_m, 0 otherwise.

5.3 LSTM model setup

We implement our LSTM model using the *lasagne* library. For the first LSTM model, we use softmax as our non-linear function and optimize the categorical cross entropy loss using adam (Kingma and Ba, 2014). For the second LSTM model, we use a sigmoid non-linear function and optimize the binary loss using adam. We use a dropout of 0.5, batch size of 100 and a learning rate of 0.001.

5.4 Results and Discussion

Table 5 shows the results of our LSTM and distant supervision based event extraction model. We compare our results with the state-of-the-art event extraction system EVEX (Hakala et al., 2013). We report the Approximate Span/Approximate Recursive metric in all our tables (described in the Shared Task (Kim et al., 2013)). The columns to the left (with column heading LSTM) show the performance of our model trained only on the official training data. The columns to the right (with column heading LSTM+Distant Supervision) show the performance of our model trained on official training data plus the additional training data of 11792 sentences we gather using our distant supervision strategy.

The table highlights some of our results. Firstly, we note that, in cases where we obtain a large number of extra sentences using distant supervision (highlighted in the column "DS Sents"), we see a considerable gain in the recall values between "LSTM" and "LSTM+Distant Supervision" models. On the contrary, in cases where we extract only a small number, we see a small gain (or sometimes even a decrease in performance). This suggests we explore further ways of selecting our extra sentences. Secondly, although the overall performance of our model using the automatic AMR parser is lower than the current state-of-the-art system, the gap of 5% in the F1 score can hopefully be reduced with the ongoing improvements in AMR parsing.

6 Related work

The biomedical event extraction task described in this work was first introduced in the BioNLP Shared Task in 2009 (Kim et al., 2009). This task helped shift the focus of relation extraction efforts from identifying simple binary interactions to identifying complex nested events that better represent the biological interactions stated frequently in text. Existing approaches to this task include SVM (Björne and Salakoski, 2013) other ML based approaches (Riedel and McCallum, 2011; Miwa et al., 2010, 2012). Methods like

	LSTM			LSTM + Distant Supervision				EVEX		
Event Type	Recall	Precision	F1	Recall	Precision	F1	DS Sents	Recall	Precision	F1
Gene_expression	**66.33**	66.55	66.44	**76.98**	61.48	68.36	**868**			
Transcription	55.10	28.57	37.63	57.14	26.92	36.60	807			
Localization	36.55	63.72	46.45	38.07	85.06	52.60	96			
Protein_catabolism	73.33	84.62	78.57	60.00	94.74	73.47	7			
==[SVT-TOTAL]==	57.82	60.86	57.27	56.35	68.05	57.60		73.83	79.56	76.59
Binding	27.61	25.94	26.75	28.57	26.12	27.29	139	41.14	44.77	42.88
Phosphorylation	**49.21**	53.75	51.38	**73.45**	45.55	56.23	**3183**			
Regulation	**16.30**	29.18	20.92	**26.07**	21.00	23.26	**2131**			
Positive_regulation	**25.98**	35.16	29.88	**37.41**	29.17	32.78	**4561**			
Negative_regulation	23.17	30.50	26.33	22.97	29.44	25.81	0			
==[REG-TOTAL]==	21.81	31.61	25.71	28.81	26.53	27.28		32.41	47.16	38.41
==[ALL-TOTAL]==	44.42	51.01	46.37	46.73	46.60	**46.66**	11792	45.44	58.03	**50.97**

Table 5: Evaluation results (Recall/Precision/F1) on the 2013 Genia Event Extraction dev set. LSTM and LSTM + Distant Supervision are our models. The last column corresponds to the results of EVEX (Hakala et al., 2013) model on the 2013 test set. Certain notable numbers are emphasized and discussed under results 5.4.

(Liu et al., 2013; MacKinlay et al., 2013) learn subgraph patterns from the event annotations in the training data and cast the event detection as subgraph matching problem. Non-feature based approaches like graph kernels compare syntactic structures directly (Airola et al., 2008; Bunescu et al., 2005). Rule based methods that either use manually crafted rules or generate rules from training data (Cohen et al., 2009; Kaljurand et al., 2009; Kilicoglu and Bergler, 2011; Bui et al., 2013) have obtained high precision on these tasks.

In our work, we take inspiration from the Turk Event Extraction System (TEES) (Björne and Salakoski, 2013) (the event extraction system for EVEX) that has consistently been the top performer in these series of tasks. They represent events using a graph format and break the event extraction task into separate multi-class classification tasks using SVM as their classifier. In our work we take a step further by making use of a deeper semantic representation as a starting point and identifying subgraphs in the AMR graph.

AMR has been successfully used for deeper semantic tasks like entity linking (Pan et al., 2015) and abstractive summarization (Mihalcea et al., 2015). Work by Garg et al. (2015) is the first one to make use of AMR representation for extracting interactions from biomedical text. They use graph kernel methods to answer the binary question of whether a given AMR subgraph expresses an interaction or not. Our work departs from theirs in that they concentrate only on binary interactions whereas we use AMR to identify complex nested events. Also, our approach additionally makes use of distant supervision to cope with the problem of limited annotated data.

Distant supervision techniques have been successfully used before for relation extraction (Mintz et al., 2009) in general domain. Recent work by (Liu et al., 2014) uses minimal supervision strategy for extracting relations particularly in biomedical texts. Our work departs from theirs in that we introduce a novel AMR path based heuristic to selectively sample the sentences obtained from distant supervision.

7 Conclusion

In this work, we show the effectiveness of using a deep semantic representation based on Abstract Meaning Representations for extracting complex nested events expressed in biomedical text. We hypothesize that an event structure is an AMR subgraph and empirically validate our hypothesis. For learning to extract such event subgraphs from AMR automatically, we develop two Recurrent Neural Network based models: one for identifying the theme, and the other for identifying the cause of the event. To overcome the dearth of manually annotated data in biomedical domain, which explains the low recall of event extraction systems, we train our model on additional training data gathered automatically using a selective distant supervision strategy. Our experiments strongly suggest that AMR parsing improvements, which are expected given the youth of this scientific field of inquiry, and the exploitation of larger, manually curated Biopax-like models and collections of biomolecular texts will be easy to capitalize on catalysts for driving future improvements in this task.

References

Antti Airola, Sampo Pyysalo, Jari Björne, Tapio Pahikkala, Filip Ginter, and Tapio Salakoski. 2008. All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC bioinformatics* 9(11):1.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Association for Computational Linguistics*.

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*. pages 16–25.

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*. Association for Computational Linguistics, pages 104–108.

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

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*. Association for Computational Linguistics, pages 50–58.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*. ACM, pages 160–167.

Emek Demir, Michael P Cary, Suzanne Paley, Ken Fukuda, Christian Lemer, Imre Vastrik, Guanming Wu, Peter D'Eustachio, Carl Schaefer, Joanne Luciano, et al. 2010. The biopax community standard for pathway data sharing. *Nature biotechnology* 28(9):935–942.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *In ACL*. Citeseer.

Sahil Garg, Aram Galstyan, Ulf Hermjakob, and Daniel Marcu. 2015. Extracting biomolecular interactions using semantic parsing of biomedical text. *arXiv preprint arXiv:1512.01587* .

Kai Hakala, Sofie Van Landeghem, Tapio Salakoski, Yves Van de Peer, and Filip Ginter. 2013. Evex in st?13: Application of a large-scale text mining resource to event extraction and network construction. In *Proceedings of the BioNLP Shared Task 2013 Workshop*. Association for Computational Linguistics, pages 26–34.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Kaarel Kaljurand, Gerold Schneider, and Fabio Rinaldi. 2009. Uzurich in the bionlp 2009 shared task. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*. Association for Computational Linguistics, pages 28–36.

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*. Association for Computational Linguistics, pages 173–182.

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 Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*. Association for Computational Linguistics, pages 1–9.

Jin-Dong Kim, Yue Wang, Nicola Colic, Seung Han Baek, Yong Hwan Kim, and Min Song. 2016. Refactoring the genia event extraction shared task toward a general framework for ie-driven kb development. *ACL 2016* page 23.

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*. Association for Computational Linguistics, pages 7–15.

Jin-Dong Kim, Yue Wang, and Yamamoto Yasunori. 2013. The genia event extraction shared task, 2013 edition-overview. In *Proceedings of the BioNLP Shared Task 2013 Workshop*. Association for Computational Linguistics, pages 8–15.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Haibin Liu, Karin Verspoor, Donald C Comeau, Andrew MacKinlay, and W John Wilbur. 2013. Generalizing an approximate subgraph matching-based system to extract events in molecular biology and cancer genetics. In *Proceedings of the BioNLP Shared Task 2013 Workshop*. pages 76–85.

Mengwen Liu, Yuan Ling, Yuan An, and Xiaohua Hu. 2014. Relation extraction from biomedical literature with minimal supervision and grouping strategy. In *Bioinformatics and Biomedicine (BIBM), 2014*

IEEE International Conference on. IEEE, pages 444–449.

Andrew MacKinlay, David Martinez, Antonio Jimeno Yepes, Haibin Liu, W John Wilbur, and Karin Verspoor. 2013. Extracting biomedical events and modifications using subgraph matching with noisy training data. In *Proceedings of the BioNLP Shared Task 2013 Workshop. Association for Computational Linguistics, Sofia, Bulgaria*. pages 35–44.

Jonathan May. 2016. Semeval-2016 task 8: Meaning representation parsing. *Proceedings of SemEval* pages 1063–1073.

Rada Mihalcea, Joyce Yue Chai, and Anoop Sarkar, editors. 2015. *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*. The Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. Association for Computational Linguistics, pages 1003–1011.

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.

Xiaoman Pan, Taylor Cassidy, Ulf Hermjakob, Heng Ji, and Kevin Knight. 2015. Unsupervised entity linking with abstract meaning representation. In (Mihalcea et al., 2015), pages 1130–1139. http://aclweb.org/anthology/N/N15/N15-1119.pdf.

Michael Pust, Ulf Hermjakob, Kevin Knight, Daniel Marcu, and Jonathan May. 2015. Using syntax-based machine translation to parse english into abstract meaning representation. In *EMNLP*.

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*. Association for Computational Linguistics, pages 46–50.

Benjamin Roth, Tassilo Barth, Michael Wiegand, and Dietrich Klakow. 2013. A survey of noise reduction methods for distant supervision. In *Proceedings of the 2013 workshop on Automated knowledge base construction*. ACM, pages 73–78.

Richard Socher, Eric H Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y Ng. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems*. pages 801–809.

Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. 2012. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, pages 721–729.

Chuan Wang, Nianwen Xue, Sameer Pradhan, and Sameer Pradhan. 2015. A transition-based algorithm for amr parsing. In *HLT-NAACL*. pages 366–375.