Cross-Sentence N-ary Relation Extraction with Graph LSTMs

Nanyun Peng[1]* **Hoifung Poon**[2] **Chris Quirk**[2] **Kristina Toutanova**[3]* **Wen-tau Yih**[2]
[1] Center for Language and Speech Processing, Computer Science Department
Johns Hopkins University, Baltimore, MD, USA
[2] Microsoft Research, Redmond, WA, USA
[3] Google Research, Seattle, WA, USA
npeng1@jhu.edu, kristout@google.com
{hoifung,chrisq,scottyih}@microsoft.com

Abstract

Past work in relation extraction has focused on binary relations in single sentences. Recent NLP inroads in high-value domains have sparked interest in the more general setting of extracting n-ary relations that span multiple sentences. In this paper, we explore a general relation extraction framework based on graph long short-term memory networks (graph LSTMs) that can be easily extended to cross-sentence n-ary relation extraction. The graph formulation provides a unified way of exploring different LSTM approaches and incorporating various intra-sentential and inter-sentential dependencies, such as sequential, syntactic, and discourse relations. A robust contextual representation is learned for the entities, which serves as input to the relation classifier. This simplifies handling of relations with arbitrary arity, and enables multi-task learning with related relations. We evaluate this framework in two important precision medicine settings, demonstrating its effectiveness with both conventional supervised learning and distant supervision. Cross-sentence extraction produced larger knowledge bases. and multi-task learning significantly improved extraction accuracy. A thorough analysis of various LSTM approaches yielded useful insight the impact of linguistic analysis on extraction accuracy.

1 Introduction

Relation extraction has made great strides in newswire and Web domains. Recently, there has been increasing interest in applying relation extraction to high-value domains such as biomedicine. The advent of $1000 human genome[1] heralds the dawn of precision medicine, but progress in personalized cancer treatment has been hindered by the arduous task of interpreting genomic data using prior knowledge. For example, given a tumor sequence, a molecular tumor board needs to determine which genes and mutations are important, and what drugs are available to treat them. Already the research literature has a wealth of relevant knowledge, and it is growing at an astonishing rate. PubMed[2], the online repository of biomedical articles, adds two new papers per minute, or one million each year. It is thus imperative to advance relation extraction for machine reading.

In the vast literature on relation extraction, past work focused primarily on binary relations in single sentences, limiting the available information. Consider the following example: "*The deletion mutation on exon-19 of* **EGFR** *gene was present in 16 patients, while the* **L858E** *point mutation on exon-21 was noted in 10. All patients were treated with* **gefitinib** *and showed a partial response.*". Collectively, the two sentences convey the fact that there is a ternary interaction between the three entities in bold, which is not expressed in either sentence alone. Namely, tumors with *L858E* mutation in *EGFR* gene can be treated with *gefitinib*. Extracting such knowledge clearly requires moving beyond binary relations and single sentences.

N-ary relations and cross-sentence extraction have received relatively little attention in the past. Prior

* This research was conducted when the authors were at Microsoft Research.

[1] http://www.illumina.com/systems/hiseq-x-sequencing-system.html

[2] https://www.ncbi.nlm.nih.gov/pubmed

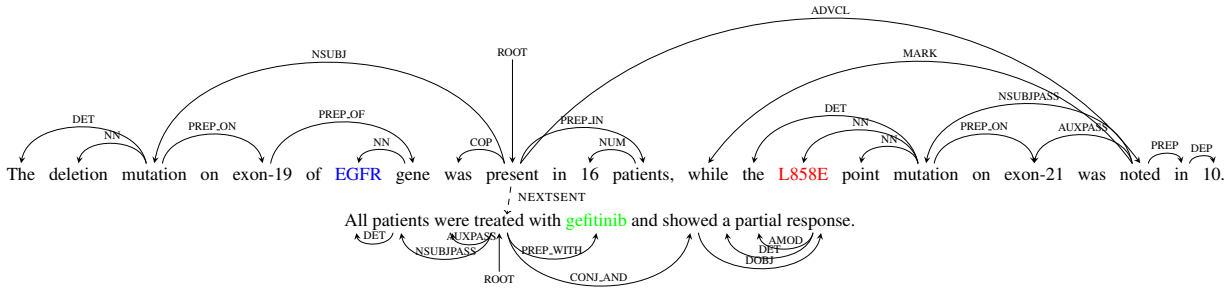

Figure 1: An example document graph for a pair of sentences expressing a ternary interaction (tumors with L858E mutation in EGFR gene respond to gefitinib treatment). For simplicity, we omit edges between adjacent words or representing discourse relations.

work on n-ary relation extraction focused on single sentences (Palmer et al., 2005; McDonald et al., 2005) or entity-centric attributes that can be extracted largely independently (Chinchor, 1998; Surdeanu and Heng, 2014). Prior work on cross-sentence extraction often used coreference to gain access to arguments in a different sentence (Gerber and Chai, 2010; Yoshikawa et al., 2011), without truly modeling inter-sentential relational patterns. (See Section 7 for a more detailed discussion.) A notable exception is Quirk and Poon (2017), which applied distant supervision to general cross-sentence relation extraction, but was limited to binary relations.

In this paper, we explore a general framework for cross-sentence n-ary relation extraction, based on graph long short-term memory networks (graph LSTMs). By adopting the graph formulation, our framework subsumes prior approaches based on chain or tree LSTMs, and can incorporate a rich set of linguistic analyses to aid relation extraction. Relation classification takes as input the entity representations learned from the entire text, and can be easily extended for arbitrary relation arity n. This approach also facilitates joint learning with kindred relations where the supervision signal is more abundant.

We conducted extensive experiments on two important domains in precision medicine. In both distant supervision and supervised learning settings, graph LSTMs that encode rich linguistic knowledge outperformed other neural network variants, as well as a well-engineered feature-based classifier. Multi-task learning with sub-relations led to further improvement. Syntactic analysis conferred a significant benefit to the performance of graph LSTMs, especially when syntax accuracy was high.

In the molecular tumor board domain, PubMed-scale extraction using distant supervision from a small set of known interactions produced orders of magnitude more knowledge, and cross-sentence extraction tripled the yield compared to single-sentence extraction. Manual evaluation verified that the accuracy is high despite the lack of annotated examples.

2 Cross-sentence n-ary relation extraction

Let $e_1, \cdots, e_m$ be entity mentions in text T. Relation extraction can be formulated as a classification problem of determining whether a relation R holds for $e_1, \cdots, e_m$ in T. For example, given a cancer patient with mutation v in gene g, a molecular tumor board seeks to find if this type of cancer would respond to drug d. Literature with such knowledge has been growing rapidly; we can help the tumor board by checking if the `Respond` relation holds for the (d, g, v) triple.

Traditional relation extraction methods focus on binary relations where all entities occur in the same sentence (i.e., $m = 2$ and T is a sentence), and cannot handle the aforementioned ternary relations. Moreover, as we focus on more complex relations and n increases, it becomes increasingly rare that the related entities will be contained entirely in a single sentence. In this paper, we generalize extraction to cross-sentence, n-ary relations, where $m > 2$ and T can contain multiple sentences. As will be shown in our experiments section, n-ary relations are crucial for high-value domains such as biomedicine, and expanding beyond the sentence boundary enables the extraction of more knowledge.

In the standard binary-relation setting, the dominant approaches are generally defined in terms of the shortest dependency path between the two entities in question, either by deriving rich features from the path or by modeling it using deep neural

networks. Generalizing this paradigm to the n-ary setting is challenging, as there are $\binom{n}{2}$ paths. One apparent solution is inspired by Davidsonian semantics: first, identify a single trigger phrase that signifies the whole relation, then reduce the n-ary relation to n binary relations between the trigger and an argument. However, challenges remain. It is often hard to specify a single trigger, as the relation is manifested by several words, often not contiguous. Moreover, it is expensive and time-consuming to annotate training examples, especially if triggers are required, as is evident in prior annotation efforts such as GENIA (Kim et al., 2009). The realistic and widely adopted paradigm is to leverage indirect supervision, such as distant supervision (Craven and Kumlien, 1999; Mintz et al., 2009), where triggers are not available.

Additionally, lexical and syntactic patterns signifying the relation will be sparse. To handle such sparsity, traditional feature-based approaches require extensive engineering and large data. Unfortunately, this challenge becomes much more severe in cross-sentence extraction when the text spans multiple sentences.

To overcome these challenges, we explore a general relation extraction framework based on graph LSTMs. By learning a continuous representation for words and entities, LSTMs can handle sparsity effectively without requiring intense feature engineering. The graph formulation subsumes prior LSTM approaches based on chains or trees, and can incorporate rich linguistic analyses.

This approach also opens up opportunities for joint learning with related relations. For example, the `Response` relation over d, g, v also implies a binary sub-relation over drug d and mutation v, with the gene underspecified. Even with distant supervision, the supervision signal for n-ary relations will likely be sparser than their binary sub-relations. Our approach makes it very easy to use multi-task learning over both the n-ary relations and their sub-relations.

3 Graph LSTMs

Learning a continuous representation can be effective for dealing with lexical and syntactic sparsity. For sequential data such as text, recurrent neural networks (RNNs) are quite popular. They resemble hidden

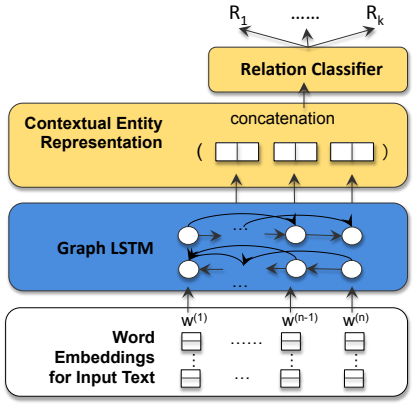

Figure 2: A general architecture for cross-sentence n-ary relation extraction based on graph LSTMs.

Markov models (HMMs), except that discrete hidden states are replaced with continuous vectors, and emission and transition probabilities with neural networks. Conventional RNNs with sigmoid units suffer from gradient diffusion or explosion, making training very difficult (Bengio et al., 1994; Pascanu et al., 2013). Long short-term memory (LSTMs) (Hochreiter and Schmidhuber, 1997) combats these problems by using a series of gates (input, forget and output) to avoid amplifying or suppressing gradients during backpropagation. Consequently, LSTMs are much more effective in capturing long-distance dependencies, and have been applied to a variety of NLP tasks. However, most approaches are based on linear chains and only explicitly model the linear context, which ignores a variety of linguistic analyses, such as syntactic and discourse dependencies.

In this section, we propose a general framework that generalizes LSTMs to graphs. While there is some prior work on learning tree LSTMs (Tai et al., 2015; Miwa and Bansal, 2016), to the best of our knowledge, graph LSTMs have not been applied to any NLP task yet. Figure 2 shows the architecture of this approach. The input layer is the word embedding of input text. Next is the graph LSTM which learns a contextual representation for each word. For the entities in question, their contextual representations are concatenated and become the input to the relation classifiers. For a multi-word entity, we simply used the average of its word representations and leave the exploration of more sophisticated aggregation approaches to future work. The layers are trained jointly with backpropagation. This framework is

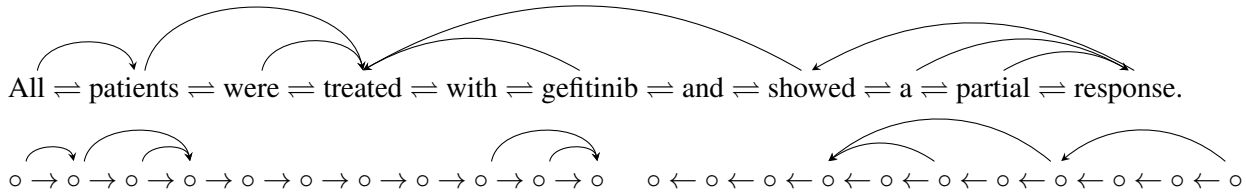

Figure 3: The graph LSTMs used in this paper. The document graph (top) is partitioned into two directed acyclic graphs (bottom); the graph LSTMs is constructed by a forward pass (Left to Right) followed by a backward pass (Right to Left). Note that information goes from dependency child to parent.

agnostic to the choice of classifiers. Jointly designing classifiers with graph LSTMs would be interesting future work.

At the core of the graph LSTM is a *document graph* that captures various dependencies among the input words. By choosing what dependencies to include in the document graph, graph LSTMs naturally subsumes linear-chain or tree LSTMs.

Compared to conventional LSTMs, the graph formulation presents new challenges. Due to potential cycles in the graph, a straightforward implementation of backpropagation might require many iterations to reach a fixed point. Moreover, in the presence of a potentially large number of edge types (adjacent-word, syntactic dependency, etc.), parametrization becomes a key problem.

In the remainder of this section, we first introduce the document graph and show how to conduct backpropagation in graph LSTMs. We then discuss two strategies for parametrizing the recurrent units. Finally, we show how to conduct multi-task learning with this framework.

3.1 Document Graph

To model various dependencies from linguistic analysis at our disposal, we follow Quirk and Poon (2017) and introduce a *document graph* to capture intra- and inter-sentential dependencies. A document graph consists of nodes that represent words and edges that represent various dependencies such as linear context (adjacent words), syntactic dependencies, and discourse relations (Lee et al., 2013; Xue et al., 2015). Figure 1 shows the document graph for our running example; this instance suggests that tumors with *L858E* mutation in *EGFR* gene responds to the drug *gefitinib*.

This document graph acts as the backbone upon which a graph LSTM is constructed. If it contains only edges between adjacent words, we recover linear-chain LSTMs. Similarly, other prior LSTM approaches can be captured in this framework by restricting edges to those in the shortest dependency path or the parse tree.

3.2 Backpropagation in Graph LSTMs

Conventional LSTMs are essentially very deep feed-forward neural networks. For example, a left-to-right linear LSTM has one hidden vector for each word. This vector is generated by a neural network (recurrent unit) that takes as input the embedding of the given word and the hidden vector of the previous word. In discriminative learning, these hidden vectors then serve as input for the end classifiers, from which gradients are backpropagated through the whole network.

Generalizing such a strategy to graphs with cycles typically requires unrolling recurrence for a number of steps (Scarselli et al., 2009; Li et al., 2016; Liang et al., 2016). Essentially, a copy of the graph is created for each step that serves as input for the next. The result is a feed-forward neural network through time, and backpropagation is conducted accordingly.

In principle, we could adopt the same strategy. Effectively, gradients are backpropagated in a manner similar to loopy belief propagation (LBP). However, this makes learning much more expensive as each update step requires multiple iterations of backpropagation. Moreover, loopy backpropagation could suffer from the same problems encountered to in LBP, such as oscillation or failure to converge.

We observe that dependencies such as coreference and discourse relations are generally sparse, so the backbone of a document graph consists of the linear chain and the syntactic dependency tree. As in belief propagation, such structures can be leveraged to make backpropagation more efficient by replac-

ing synchronous updates, as in the unrolling strategy, with asynchronous updates, as in linear-chain LSTMs. This opens up opportunities for a variety of strategies in ordering backpropagation updates.

In this paper, we adopt a simple strategy that performed quite well in preliminary experiments, and leave further exploration to future work. Specifically, we partition the document graph into two directed acyclic graphs (DAGs). One DAG contains the left-to-right linear chain, as well as other forward-pointing dependencies. The other DAG covers the right-to-left linear chain and the backward-pointing dependencies. Figure 3 illustrates this strategy. Effectively, we partition the original graph into the forward pass (left-to-right), followed by the backward pass (right-to-left), and construct the LSTMs accordingly. When the document graph only contains linear chain edges, the graph LSTMs is exactly a bi-directional LSTMs (BiLSTMs).

3.3 The Basic Recurrent Propagation Unit

A standard LSTM unit consists of an input vector (word embedding), a memory cell and an output vector (contextual representation), as well as several gates. The *input gate* and *output gate* control the information flowing into and out of the cell, whereas the *forget gate* can optionally remove information from the recurrent connection to a precedent unit.

In linear-chain LSTMs, each unit contains only one forget gate, as it has only one direct precedent (i.e., the adjacent-word edge pointing to the previous word). In graph LSTMs, however, a unit may have several precedents, including connections to the same word via different edges. We thus introduce a forget gate for each precedent, similar to the approach taken by Tai et al. (2015) for tree LSTMs.

Encoding rich linguistic analysis introduces many distinct edge types besides word adjacency, such as syntactic dependencies, which opens up many possibilities for parametrization. This was not considered in prior syntax-aware LSTM approaches (Tai et al., 2015; Miwa and Bansal, 2016). In this paper, we explore two schemes that introduce more fined-grained parameters based on the edge types.

Full Parametrization Our first proposal simply introduces a different set of parameters for each edge type, with computation specified below.

$$
\begin{aligned}
i_t &= \sigma(W_i x_t + \sum\nolimits_{j \in P(t)} U_i^{m(t,j)} h_j + b_i) \\
o_t &= \sigma(W_o x_t + \sum\nolimits_{j \in P(t)} U_o^{m(t,j)} h_j + b_o) \\
\tilde{c}_t &= \tanh(W_c x_t + \sum\nolimits_{j \in P(t)} U_c^{m(t,j)} h_j + b_c) \\
f_{tj} &= \sigma(W_f x_t + U_f^{m(t,j)} h_j + b_f) \\
c_t &= i_t \odot \tilde{c}_t + \sum\nolimits_{j \in P(t)} f_{tj} \odot c_j \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

As in standard chain LSTMs, x_t is the input word vector for node t, h_t is the hidden state vector for node t, W's are the input weight matrices, and b's are the bias vectors. σ, $\tanh$, and $\odot$ represent the sigmoid function, the hyperbolic tangent function, and the Hadamard product (pointwise multiplication), respectively. The main differences lie in the recurrence terms. In graph LSTMs, a unit might have multiple predecessors ($P(t)$), for each of which (j) there is a forget gate f_{tj}, and a typed weight matrix $U^{m(t,j)}$, where $m(t, j)$ signifies the connection type between t and j. The *input* and *output* gates (i_t, o_t) depend on all predecessors, whereas the forget gate (f_{tj}) only depends on the predecessor with which the gate is associated. c_t and $\tilde{c}_t$ represent intermediate computation results within the memory cell, which take into account the input and forget gates, and will be combined with output gate to produce the hidden representation h_t.

Full parameterization is straightforward, but it requires a large number of parameters when there are many edge types. For example, there are dozens of syntactic edge types, each corresponding to a Stanford dependency label. As a result, in our experiments we resort to using only the coarse-grained types: word adjacency, syntactic dependency, etc. Next, we will consider a more fine-grained approach by learning an edge-type embedding.

Edge-Type Embedding To reduce the number of parameters and leverage potential correlation among fine-grained edge types, we learned a low-dimensional embedding of the edge types, and conducted an outer product of the predecessor's hidden vector and the edge-type embedding to generate a "typed hidden representation", which is a matrix. The new computation is as follows:

$$
\begin{aligned}
i_t &= \sigma(W_i x_t + \sum_{j \in P(t)} U_i \times_T (h_j \otimes e_j) + b_i) \\
f_{tj} &= \sigma(W_f x_t + U_f \times_T (h_j \otimes e_j) + b_f) \\
o_t &= \sigma(W_o x_t + \sum_{j \in P(t)} U_o \times_T (h_j \otimes e_j) + b_o) \\
\tilde{c}_t &= \tanh(W_c x_t + \sum_{j \in P(t)} U_c \times_T (h_j \otimes e_j) + b_c) \\
c_t &= i_t \odot \tilde{c}_t + \sum_{j \in P(t)} f_{tj} \odot c_j \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

U's are now $l \times l \times d$ tensors (l is the dimension of the hidden vector and d is the dimension for edge-type embedding), and $h_j \otimes e_j$ is a tensor product that produces an $l \times d$ matrix. $\times_T$ denotes a tensor dot product defined as $T \times_T A = \sum_d (T_{:,:,d} \cdot A_{:,d})$, which produces an l-dimensional vector. The edge-type embedding e_j is jointly trained with the other parameters.

3.4 Comparison with Prior LSTM Approaches

The main advantages of a graph formulation are its generality and flexibility. As seen in Section 3.1, linear-chain LSTMs are a special case when the document graph is the linear chain of adjacent words. Similarly, Tree LSTMs (Tai et al., 2015) are a special case when the document graph is the parse tree.

In graph LSTMs, the encoding of linguistic knowledge is factored from the backpropagation strategy (Section 3.2), making it much more flexible, including introducing cycles. For example, Miwa and Bansal (2016) conducted joint entity and binary relation extraction by stacking a LSTM for relation extraction on top of another LSTM for entity recognition. In graph LSTMs, the two can be combined seamlessly using a document graph comprising both the word-adjacency chain and the dependency path between the two entities.

The document graph can also incorporate other linguistic information. For example, coreference and discourse parsing are intuitively relevant for cross-sentence relation extraction. Although existing systems have not yet been shown to improve cross-sentence relation extraction (Quirk and Poon, 2017), it remains an important future direction to explore incorporating such analyses, especially after adapting them to the biomedical domains (Bell et al., 2016).

3.5 Multi-task Learning with Sub-relations

Multi-task learning has been shown to be beneficial in training neural networks (Caruana, 1998; Collobert and Weston, 2008; Peng and Dredze, 2016). By learning contextual entity representations, our framework makes it straightforward to conduct multi-task learning. The only change is to add a separate classifier for each related auxiliary relation. All classifiers share the same graph LSTMs representation learner and word embeddings, and can potentially help each other by pooling their supervision signals.

In the molecular tumor board domain, we applied this paradigm to joint learning of both the ternary relation (drug-gene-mutation) and its binary sub-relation (drug-mutation). Experiment results show that this provides significant gains in both tasks.

4 Implementation Details

We implemented our methods using the Theano library (Theano Development Team, 2016). We used logistic regression for our relation classifiers. Hyper parameters were set based on preliminary experiments on a small development dataset. Training was done using mini-batched stochastic gradient descent (SGD) with batch size 8. We used a learning rate of 0.02 and trained for at most 30 epochs, with early stopping based on development data (Caruana et al., 2001; Graves et al., 2013). The dimension for the hidden vectors in LSTM units was set to 150, and the dimension for the edge-type embedding was set to 3. The word embeddings were initialized with the publicly available 100-dimensional GloVe word vectors trained on 6 billion words from Wikipedia and web text[3] (Pennington et al., 2014). Other model parameters were initialized with random samples drawn uniformly from the range $[-1, 1]$.

In multi-task training, we alternated among all tasks, each time passing through all data for one task[4], and updating the parameters accordingly. This was repeated for 30 epochs.

[3] `http://nlp.stanford.edu/projects/glove/`

[4] However, drug-gene pairs have much more data, so we subsampled the instances down to the same size as the main n-ary relation task.

5 Domain: Molecular Tumor Boards

Our main experiments focus on extracting ternary interactions over drugs, genes and mutations, which is important for molecular tumor boards. A drug-gene-mutation interaction is broadly construed as an association between the drug efficacy and the mutation in the given gene. There is no annotated dataset for this problem. However, due to the importance of such knowledge, oncologists have been painstakingly curating known relations from reading papers. Such a manual approach cannot keep up with the rapid growth of the research literature, and the coverage is generally sparse and not up to date. However, the curated knowledge can be used for distant supervision.

5.1 Datasets

We obtained biomedical literature from PubMed Central[5], consisting of approximately one million full-text articles as of 2015. Note that only a fraction of papers contain knowledge about drug-gene-mutation interactions. Extracting such knowledge from the vast body of biomedical papers is exactly the challenge. As we will see in later subsections, distant supervision enables us to generate a sizable training set from a small number of manually curated facts, and the learned model was able to extract orders of magnitude more facts. In future work, we will explore incorporating more known facts for distant supervision and extracting from more full-text articles.

We conducted tokenization, part-of-speech tagging, and syntactic parsing using SPLAT (Quirk et al., 2012), and obtained Stanford dependencies (de Marneffe et al., 2006) using Stanford CoreNLP (Manning et al., 2014). We used the entity taggers from Literome (Poon et al., 2014) to identify drug, gene and mutation mentions.

We used the Gene Drug Knowledge Database (GDKD) (Dienstmann et al., 2015) and the Clinical Interpretations of Variants In Cancer (CIVIC) knowledge base[6] for distant supervision. The knowledge bases distinguish fine-grained interaction types, which we do not use in this paper.

[5] `http://www.ncbi.nlm.nih.gov/pmc/`

[6] `http://civic.genome.wustl.edu`

5.2 Distant Supervision

After identifying drug, gene and mutation mentions in the text, co-occurring triples with known interactions were chosen as positive examples. However, unlike the single-sentence setting in standard distant supervision, care must be taken in selecting the candidates. Since the triples can reside in different sentences, an unrestricted selection of text spans would risk introducing many obviously wrong examples. We thus followed Quirk and Poon (2017) in restricting the candidates to those occurring in a *minimal span*, i.e., we retain a candidate only if is no other co-occurrence of the same entities in an overlapping text span with a smaller number of consecutive sentences. Furthermore, we avoid picking unlikely candidates where the triples are far apart in the document. Specifically, we considered entity triples within K consecutive sentences, ignoring paragraph boundaries. $K = 1$ corresponds to the baseline of extraction within single sentences. We explored $K \leq 3$, which captured a large fraction of candidates without introducing many unlikely ones.

Only 59 distinct drug-gene-mutation triples from the knowledge bases were matched in the text. Even from such a small set of unique triples, we obtained 3,462 ternary relation instances that can serve as positive examples. For multi-task learning, we also considered drug-gene and drug-mutation sub-relations, which yielded 137,469 drug-gene and 3,192 drug-mutation relation instances as positive examples.

We generate negative examples by randomly sampling co-occurring entity triples without known interactions, subject to the same restrictions above. We sampled the same number as positive examples to obtain a balanced dataset[7].

5.3 Automatic Evaluation

To compare the various models in our proposed framework, we conducted five-fold cross-validation, treating the positive and negative examples from distant supervision as gold annotation. To avoid train-test contamination, all examples from a document were assigned to the same fold. Since our datasets are balanced by construction, we simply report average test accuracy on held-out folds. Obviously, the

[7] We will release the dataset at `http://hanover.azurewebsites.net`.

Model	Single-Sent.	Cross-Sent.
Feature-Based	74.7	77.7
CNN	77.5	78.1
BiLSTM	75.3	80.1
Graph LSTM - EMBED	76.5	80.6
Graph LSTM - FULL	**77.9**	**80.7**

Table 1: Average test accuracy in five-fold cross-validation for drug-gene-mutation ternary interactions. Feature-Based used the best performing model in (Quirk and Poon, 2017) with features derived from shortest paths between all entity pairs.

Model	Single-Sent.	Cross-Sent.
Feature-Based	73.9	75.2
CNN	73.0	74.9
BiLSTM	73.9	76.0
BiLSTM-Shortest-Path	70.2	71.7
Tree LSTM	**75.9**	75.9
Graph LSTM-EMBED	74.3	76.5
Graph LSTM-FULL	75.6	**76.7**

Table 2: Average test accuracy in five-fold cross-validation for drug-mutation binary relations, with an extra baseline using a BiLSTM on the shortest dependency path (Xu et al., 2015b; Miwa and Bansal, 2016).

results could be noisy (e.g., entity triples not known to have an interaction might actually have one), but this evaluation is automatic and can quickly evaluate the impact of various design choices.

We evaluated two variants of graph LSTMs: "Graph LSTM-FULL" with full parametrization and "Graph LSTM-EMBED" with edge-type embedding. We compared graph LSTMs with three strong baseline systems: a well-engineered feature-based classifier (Quirk and Poon, 2017), a convolutional neural network (CNN) (Zeng et al., 2014; Santos et al., 2015; Wang et al., 2016), and a bi-directional LSTM (BiLSTM). Following Wang et al. (2016), we used input attention for the CNN and a input window size of 5. Quirk and Poon (2017) only extracted binary relations. We extended it to ternary relations by deriving features for each entity pair (with added annotation to signify the two entity types), and pooling the features from all pairs.

For binary relation extraction, prior syntax-aware approaches are directly applicable. So we also compared with a state-of-the-art tree LSTM system (Miwa and Bansal, 2016) and a BiLSTM on the shortest dependency path between the two entities (BiLSTM-Shortest-Path) (Xu et al., 2015b).

Table 1 shows the results for cross-sentence, ternary relation extraction. All neural-network based models outperformed the feature-based classifier, illustrating their advantage in handling sparse linguistic patterns without requiring intense feature engineering. All LSTMs significantly outperformed CNN in the cross-sentence setting, verifying the importance in capturing long-distance dependencies.

The two variants of graph LSTMs perform on par with each other, though Graph LSTM-FULL has a small advantage, suggesting that further exploration of parametrization schemes could be beneficial. In particular, the edge-type embedding might improve by pretraining on unlabeled text with syntactic parses.

Both graph variants significantly outperformed BiLSTMs ($p < 0.05$ by McNemar's chi-square test), though the difference is small. This result is intriguing. In Quirk and Poon (2017), the best system incorporated syntactic dependencies and outperformed the linear-chain variant (Base) by a large margin. So why didn't graph LSTMs make an equally substantial gain by modeling syntactic dependencies?

One reason is that linear-chain LSTMs can already captured some of the long-distance dependencies available in syntactic parses. BiLSTMs substantially outperformed the feature-based classifier, even without explicit modeling of syntactic dependencies. The gain cannot be entirely attributed to word embedding as LSTMs also outperformed CNNs.

Another reason is that syntactic parsing is less accurate in the biomedical domain. Parse errors confuse the graph LSM learner, limiting the potential for gain. In Section 6, we show supporting evidence in a domain when gold parses are available.

We also reported accuracy on instances within single sentences, which exhibited a broadly similar set of trends. Note that single-sentence and cross-sentence accuracies are not directly comparable, as the test sets are different (one subsumes the other).

We conducted the same experiments on the binary sub-relation between drug-mutation pairs. Table 2

	Drug-Gene-Mut.	Drug-Mut.
BiLSTM	80.1	76.0
+Multi-task	**82.4**	78.1
Graph LSTM	80.7	76.7
+Multi-task	82.0	**78.5**

Table 3: Multi-task learning improved accuracy for both BiLSTMs and Graph LSTMs.

	Single-Sent.	Cross-Sent.
Candidates	10,873	57,033
$p \geq 0.5$	1,408	4,279
$p \geq 0.9$	530	1,461
GDKD + CIVIC	59	

Table 4: Numbers of unique drug-gene-mutation interactions extracted from PubMed Central articles, compared to that from manually curated KBs used in distant supervision. p signifies output probability.

shows the results, which are similar to the ternary case: Graph LSTM-FULL consistently performed the best for both single sentence and cross-sentence instances. BiLSTMs on the shortest path substantially underperformed BiLSTMs or graph LSTMs, losing between 4-5 absolute points in accuracy, which could be attributed to the lower parsing quality in the biomedical domain. Interestingly, the state-of-the-art tree LSTMs (Miwa and Bansal, 2016) also underperformed graph LSTMs, even though they encoded essentially the same linguistic structures (word adjacency and syntactic dependency). We attributed the gain to the fact that Miwa and Bansal (2016) used separate LSTMs for the linear chain and the dependency tree, whereas graph LSTMs learned a single representation for both.

To evaluate whether joint learning with sub-relations can help, we conducted multi-task learning using Graph LSTM-FULL to jointly train extractors for both the ternary interaction and the drug-mutation, drug-gene sub-relations. Table 3 shows the results. Multi-task learning resulted in a significant gain for both the ternary interaction and the drug-mutation interaction. Interestingly, the advantage of graph LSTMs over BiLSTMs is reduced with multi-task learning, suggesting that with more supervision signal, even linear-chain LSTMs can learn to capture long-range dependencies that are were made evident by parse features in graph LSTMs. Note that there are many more instances for drug-gene interaction than others, so we only sampled a subset of comparable size. Therefore, we do not evaluate the performance gain for drug-gene interaction, as in practice, one would simply learn from all available data, and the sub-sampled results are not competitive.

We included coreference and discourse relations in our document graph. However, we didn't observe any significant gains, similar to the observation in Quirk and Poon (2017). We leave further exploration to future work.

5.4 PubMed-Scale Extraction

Our ultimate goal is to extract all knowledge from available text. We thus retrained our model using the best system from automatic evaluation (i.e., Graph LSTM-FULL) on all available data. The resulting model was then used to extract relations from all PubMed Central articles.

Table 4 shows the number of candidates and extracted interactions. With as little as 59 unique drug-gene-mutation triples from the two databases[8], we learned to extract orders of magnitude more unique interactions. The results also highlight the benefit of cross-sentence extraction, which yields 3 to 5 times more relations than single-sentence extraction.

Table 5 conducts a similar comparison on unique number of drugs, genes, and mutations. Again, machine reading covers far more unique entities, especially with cross-sentence extraction.

5.5 Manual Evaluation

Our automatic evaluations are useful for comparing competing approaches, but may not reflect the true classifier precision as the labels are noisy. Therefore, we randomly sampled extracted relation instances and asked three researchers knowledgeable in precision medicine to evaluate their correctness. For each instance, the annotators were presented with the provenance: sentences with the drug, gene, and mutation highlighted. The annotators determined in

[8]There are more in the databases, but these are the only ones for which we found matching instances in the text. In future work, we will explore various ways to increase the number, e.g., by matching underspecified drug classes to specific drugs.

	Drug	Gene	Mut.
GDKD + CIVIC	16	12	41
Single-Sent. ($p \geq 0.9$)	68	228	221
Single-Sent. ($p \geq 0.5$)	93	597	476
Cross-Sent. ($p \geq 0.9$)	103	512	445
Cross-Sent. ($p \geq 0.5$)	144	1344	1042

Table 5: Numbers of unique drugs, genes and mutations in extraction from PubMed Central articles, in comparison with that in the manually curated Gene Drug Knowledge Database (GDKD) and Clinical Interpretations of Variants In Cancer (CIVIC) used for distant supervision. p signifies output probability.

	Precision	Entity Error	Relation Error
Random	17%	36%	47%
$p \geq 0.5$	64%	7%	29%
$p \geq 0.9$	75%	1%	24%

Table 6: Sample precision of drug-gene-mutation interactions extracted from PubMed Central articles. p signifies output probability.

each case whether this instance implied that the given entities were related. Note that evaluation does not attempt to identify whether the relationships are true or replicated in follow-up papers; rather, it focuses on whether the relationships are entailed by the text.

We focused our evaluation efforts on the cross-sentence ternary-relation setting. We considered three probability thresholds: 0.9 for a high-precision but potentially low-recall setting, 0.5, and a random sample of all candidates. In each case, 150 instances were selected for a total of 450 annotations. A subset of 150 instances were reviewed by two annotators, and the inter-annotator agreement was 88%.

Table 6 shows that the classifier indeed filters out a large portion of potential candidates, with estimated instance accuracy of 64% at the threshold of 0.5, and 75% at 0.9. Interestingly, LSTMs are effective at screening out many entity mention errors, presumably because they include broad contextual features.

Model	Precision	Recall	F1
Poon et al. (2015)	37.5	29.9	33.2
BiLSTM	37.6	29.4	33.0
Graph LSTM	41.4	30.0	34.8
Graph LSTM (GOLD)	**43.3**	**30.5**	**35.8**

Table 7: GENIA test results on the binary relation of gene regulation. Graph LSTM (GOLD) used gold syntactic parses in the document graph.

6 Domain: Genetic Pathways

We also conducted experiments on extracting genetic pathway interactions using the GENIA Event Extraction dataset (Kim et al., 2009). This dataset contains gold syntactic parses for the sentences, which offered a unique opportunity to investigate the impact of syntactic analysis on graph LSTMs. It also allowed us to test our framework in supervised learning.

The original shared task evaluated on complex, nested events for nine event types, many of which are unary relations (Kim et al., 2009). Following Poon et al. (2015), we focused on gene regulation and reduced it to binary-relation classification for head-to-head comparison. We followed their experimental protocol by sub-sampling negative examples to be about three times of positive examples.

Since the dataset is not entirely balanced, we reported precision, recall, and F1. We used our best performing graph LSTM from the previous experiments. By default, automatic parses were used in the document graphs, whereas in Graph LSTM (GOLD), gold parses were used instead. Table 7 shows the results. Once again, despite the lack of intense feature engineering, linear-chain LSTMs performed on par with the feature-based classifier (Poon et al., 2015). Graph LSTMs exhibited a more commanding advantage over linear-chain LSTMs in this domain, substantially outperforming the latter ($p < 0.01$ by McNemar's chi-square test). Most interestingly, graph LSTMs using gold parses significantly outperformed that using automatic parses, suggesting that encoding high-quality analysis is particularly beneficial.

7 Related Work

Most work on relation extraction has been applied to binary relations of entities in a single sentence. We first review relevant work on the single-sentence bi-

nary relation extraction task, and then review related work on n-ary and cross-sentence relation extraction.

Binary relation extraction The traditional feature-based methods rely on carefully designed features to learn good models, and often integrate diverse sources of evidence such as word sequences and syntax context (Kambhatla, 2004; GuoDong et al., 2005; Boschee et al., 2005; Suchanek et al., 2006; Chan and Roth, 2010; Nguyen and Grishman, 2014). The kernel-based methods design various subsequence or tree kernels (Mooney and Bunescu, 2005; Bunescu and Mooney, 2005; Qian et al., 2008) to capture structured information. Recently, models based on neural networks have advanced the state of the art by automatically learning powerful feature representations (Xu et al., 2015a; Zhang et al., 2015; Santos et al., 2015; Xu et al., 2015b; Xu et al., 2016).

Most neural architectures resemble Figure 2, where there is a core representation learner (blue) that takes word embeddings as input and produces contextual entity representations. Such representations are then taken by relation classifiers to produce the final predictions. Effectively representing sequences of words, both convolutional (Zeng et al., 2014; Wang et al., 2016; Santos et al., 2015) and RNN-based architectures (Zhang et al., 2015; Socher et al., 2012; Cai et al., 2016) have been successful. Most of these have focused on modeling either the surface word sequences or the hierarchical syntactic structure. Miwa and Bansal (2016) proposed an architecture that benefits from both types of information, using a surface sequence layer, followed by a dependency-tree sequence layer.

N-ary relation extraction Early work on extracting relations between more than two arguments has been done in MUC-7, with a focus on fact/event extraction from news articles (Chinchor, 1998). Semantic role labeling in the Propbank (Palmer et al., 2005) or FrameNet (Baker et al., 1998) style are also instances of n-ary relation extraction, with extraction of events expressed in a single sentence. McDonald et al. (2005) extract n-ary relations in a biomedical domain, by first factoring the n-ary relation into pair-wise relations between all entity pairs, and then constructing maximal cliques of related entities. Recently, neural models have been applied to semantic role labeling (FitzGerald et al., 2015; Roth and Lapata, 2016). These works learned neural representations by effectively decomposing the n-ary relation into binary relations between the predicate and each argument, by embedding the dependency path between each pair, or by combining features of the two using a feed-forward network. Although some re-ranking or joint inference models have been employed, the representations of the individual arguments do not influence each other. In contrast, we propose a neural architecture that jointly represents n entity mentions, taking into account long-distance dependencies and inter-sentential information.

Cross-sentence relation extraction Several relation extraction tasks have benefited from cross-sentence extraction, including MUC fact and event extraction (Swampillai and Stevenson, 2011), record extraction from web pages (Wick et al., 2006), extraction of facts for biomedical domains (Yoshikawa et al., 2011), and extensions of semantic role labeling to cover implicit inter-sentential arguments (Gerber and Chai, 2010). These prior works have either relied on explicit co-reference annotation, or on the assumption that the whole document refers to a single coherent event, to simplify the problem and reduce the need for powerful representations of multi-sentential contexts of entity mentions. Recently, cross-sentence relation extraction models have been learned with distant supervision, and used integrated contextual evidence of diverse types without reliance on these assumptions (Quirk and Poon, 2017), but that work focused on binary relations only and explicitly engineered sparse indicator features.

Relation extraction using distant supervision Distant supervision has been applied to extraction of binary (Mintz et al., 2009; Poon et al., 2015) and n-ary (Reschke et al., 2014; Li et al., 2015) relations, traditionally using hand-engineered features. Neural architectures have recently been applied to distantly supervised extraction of binary relations (Zeng et al., 2015). Our work is the first to propose a neural architecture for n-ary relation extraction, where the representation of a tuple of entities is not decomposable into independent representations of the individual entities or entity pairs, and which integrates diverse information from multi-sentential context. To utilize training data more effectively, we show how multi-task learning for component binary sub-relations can

improve performance. Our learned representation combines information sources within a single sentence in a more integrated and generalizable fashion than prior approaches, and can also improve performance on single-sentence binary relation extraction.

8 Conclusion

We explore a general framework for cross-sentence n-ary relation extraction based on graph LSTMs. The graph formulation subsumes linear-chain and tree LSTMs and makes it easy to incorporate rich linguistic analysis. Experiments on biomedical domains showed that extraction beyond the sentence boundary produced far more knowledge, and encoding rich linguistic knowledge provided consistent gain.

While there is much room to improve in both recall and precision, our results indicate that machine reading can already be useful in precision medicine. In particular, automatically extracted facts (Section 5.4) can serve as candidates for manual curation. Instead of scanning millions of articles to curate from scratch, human curators would just quickly vet thousands of extractions. The errors identified by curators offer direct supervision to the machine reading system for continuous improvement. Therefore, the most important goal is to attain high recall and reasonable precision. Our current models are already quite capable.

Future directions include: interactive learning with user feedback; improving discourse modeling in graph LSTMs; exploring other backpropagation strategies; joint learning with entity linking; applications to other domains.

Acknowledgements

We thank Daniel Fried and Ming-Wei Chang for useful discussions, as well as the anonymous reviewers and editor-in-chief Mark Johnson for their helpful comments.

References

Collin Baker, Charles Fillmore, and John Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics*.

Dane Bell, Gustave Hahn-Powell, Marco A. Valenzuela-Escarcega, and Mihai Surdeanu. 2016. An investigation of coreference phenomena in the biomedical domain. In *Proceedings of the Tenth Edition of the Language Resources and Evaluation Conference*.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2).

Elizabeth Boschee, Ralph Weischedel, and Alex Zamanian. 2005. Automatic information extraction. In *Proceedings of the International Conference on Intelligence Analysis*.

Razvan C Bunescu and Raymond J Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Rui Cai, Xiaodong Zhang, and Houfeng Wang. 2016. Bidirectional recurrent convolutional neural network for relation classification. In *Proceedings of the Fifty-Fourth Annual Meeting of the Association for Computational Linguistics*.

Rich Caruana, Steve Lawrence, and Lee Giles. 2001. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Proceedings of The Fifteenth Annual Conference on Neural Information Processing Systems*.

Rich Caruana. 1998. Multitask learning. In *Learning to learn*. Springer.

Yee Seng Chan and Dan Roth. 2010. Exploiting background knowledge for relation extraction. In *Proceedings of the Twenty-Third International Conference on Computational Linguistics*.

Nancy Chinchor. 1998. Overview of MUC-7/MET-2. Technical report, Science Applications International Corporation, San Diego, CA.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the Twenty-Fifth International Conference on Machine learning*.

Mark Craven and Johan Kumlien. 1999. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*.

Rodrigo Dienstmann, In Sock Jang, Brian Bot, Stephen Friend, and Justin Guinney. 2015. Database of genomic biomarkers for cancer drugs and clinical targetability in solid tumors. *Cancer Discovery*, 5.

Nicholas FitzGerald, Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. 2015. Semantic role labeling with neural network factors. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Matthew Gerber and Joyce Y. Chai. 2010. Beyond NomBank: A study of implicit arguments for nominal predicates. In *Proceedings of the Forty-Eighth Annual Meeting of the Association for Computational Linguistics*.

Alan Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Proceedings of The Thirty-Eighth IEEE International Conference on Acoustics, Speech and Signal Processing*.

Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the Forty-Third Annual Meeting of the Association for Computational Linguistics*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8).

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the Forty-Second Annual Meeting of the Association for Computational Linguistics, Demonstration Sessions*.

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*.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4).

Hong Li, Sebastian Krause, Feiyu Xu, Andrea Moro, Hans Uszkoreit, and Roberto Navigli. 2015. Improvement of n-ary relation extraction by adding lexical semantics to distant-supervision rule learning. In *Proceedings of the Seventh International Conference on Agents and Artificial Intelligence*.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. 2016. Gated graph sequence neural networks. In *Proceedings of the Fourth International Conference on Learning Representations*.

Xiaodan Liang, Xiaohui Shen, Jiashi Feng, Liang Lin, and Shuicheng Yan. 2016. Semantic object parsing with graph LSTM. In *Proceedings of European Conference on Computer Vision*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the Fifty-Second Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Ryan McDonald, Fernando Pereira, Seth Kulick, Scott Winters, Yang Jin, and Pete White. 2005. Simple algorithms for complex relation extraction with applications to biomedical IE. In *Proceedings of the Forty-Third Annual Meeting on Association for Computational Linguistics*.

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 Forty-Seventh Annual Meeting of the Association for Computational Linguistics and the Fourth International Joint Conference on Natural Language Processing*.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the Fifty-Fourth Annual Meeting of the Association for Computational Linguistics*.

Raymond J Mooney and Razvan C Bunescu. 2005. Subsequence kernels for relation extraction. In *Proceedings of The Nineteen Annual Conference on Neural Information Processing Systems*.

Thien Huu Nguyen and Ralph Grishman. 2014. Employing word representations and regularization for domain adaptation of relation extraction. In *Proceedings of the Fifty-Second Annual Meeting of the Association for Computational Linguistics*.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1).

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of The Thirtieth International Conference on Machine Learning*.

Nanyun Peng and Mark Dredze. 2016. Improving named entity recognition for chinese social media with word segmentation representation learning. In *Proceedings of the Fifty-Fourth Annual Meeting of the Association for Computational Linguistics*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Hoifung Poon, Chris Quirk, Charlie DeZiel, and David Heckerman. 2014. Literome: PubMed-scale genomic knowledge base in the cloud. *Bioinformatics*, 30(19).

Hoifung Poon, Kristina Toutanova, and Chris Quirk. 2015. Distant supervision for cancer pathway extraction from text. In *Pacific Symposium on Biocomputing*.

Longhua Qian, Guodong Zhou, Fang Kong, Qiaoming Zhu, and Peide Qian. 2008. Exploiting constituent

dependencies for tree kernel-based semantic relation extraction. In *Proceedings of the Twenty-Second International Conference on Computational Linguistics*.

Chris Quirk and Hoifung Poon. 2017. Distant supervision for relation extraction beyond the sentence boundary. In *Proceedings of the Fifteenth Conference on European chapter of the Association for Computational Linguistics*.

Chris Quirk, Pallavi Choudhury, Jianfeng Gao, Hisami Suzuki, Kristina Toutanova, Michael Gamon, Wen-tau Yih, and Lucy Vanderwende. 2012. MSR SPLAT, a language analysis toolkit. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Demonstration Session*.

Kevin Reschke, Martin Jankowiak, Mihai Surdeanu, Christopher D Manning, and Daniel Jurafsky. 2014. Event extraction using distant supervision. In *Proceedings of Eighth edition of the Language Resources and Evaluation Conference*.

Michael Roth and Mirella Lapata. 2016. Neural semantic role labeling with dependency path embeddings. In *Proceedings of the Fifty-Fourth Annual Meeting of the Association for Computational Linguistics*.

Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of the Fifty-Third Annual Meeting of the Association for Computational Linguistics*.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1).

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Fabian M Suchanek, Georgiana Ifrim, and Gerhard Weikum. 2006. Combining linguistic and statistical analysis to extract relations from web documents. In *Proceedings of the Twelfth International Conference on Knowledge Discovery and Data Mining*.

Mihai Surdeanu and Ji Heng. 2014. Overview of the english slot filling track at the TAC2014 knowledge base population evaluation. In *Proceedings of the U.S. National Institute of Standards and Technology Knowledge Base Population 2014 Workshop*.

Kumutha Swampillai and Mark Stevenson. 2011. Extracting relations within and across sentences. In *Proceedings of the Conference on Recent Advances in Natural Language Processing*.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the Fifty-Third Annual Meeting of the Association for Computational Linguistics*.

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688.

Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. 2016. Relation classification via multi-level attention CNNs. In *Proceedings of the Fifty-Fourth Annual Meeting of the Association for Computational Linguistics*.

Michael Wick, Aron Culotta, and Andrew McCallum. 2006. Learning field compatibilities to extract database records from unstructured text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2015a. Semantic relation classification via convolutional neural networks with simple negative sampling. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015b. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*.

Yan Xu, Ran Jia, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, and Zhi Jin. 2016. Improved relation classification by deep recurrent neural networks with data augmentation. In *Proceedings of the Twenty-Sixth International Conference on Computational Linguistics*.

Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Rashmi Prasad, Christopher Bryant, and Attapol Rutherford. 2015. The CoNLL-2015 shared task on shallow discourse parsing. In *Proceedings of the Conference on Computational Natural Language Learning, Shared Task*.

Katsumasa Yoshikawa, Sebastian Riedel, Tsutomu Hirao, Masayuki Asahara, and Yuji Matsumoto. 2011. Coreference based event-argument relation extraction on biomedical text. *Journal of Biomedical Semantics*, 2(5).

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. 2014. Relation classification via convolutional deep neural network. In *Proceedings of the Twenty-Sixth International Conference on Computational Linguistics*.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. 2015. Bidirectional long short-term memory networks for relation classification. In *Proceedings of Twenty-Ninth Pacific Asia Conference on Language, Information and Computation*.