Entity, Relation, and Event Extraction with Contextualized Span Representations

David Wadden[†] Ulme Wennberg[†] Yi Luan[‡] Hannaneh Hajishirzi[†*]
[†]Paul G. Allen School of Computer Science & Engineering, University of Washington
[‡]Google AI Language [*]Allen Institute for Artificial Intelligence
{dwadden,ulme,hannaneh}@cs.washington.edu
luanyi@google.com

Abstract

We examine the capabilities of a unified, multi-task framework for three information extraction tasks: named entity recognition, relation extraction, and event extraction. Our framework (called DYGIE++) accomplishes all tasks by enumerating, refining, and scoring text spans designed to capture local (within-sentence) and global (cross-sentence) context. Our framework achieves state-of-the-art results across all tasks, on four datasets from a variety of domains. We perform experiments comparing different techniques to construct span representations. Contextualized embeddings like BERT perform well at capturing relationships among entities in the same or adjacent sentences, while dynamic span graph updates model long-range cross-sentence relationships. For instance, propagating span representations via predicted coreference links can enable the model to disambiguate challenging entity mentions. Our code is publicly available at https://github.com/dwadden/dygiepp and can be easily adapted for new tasks or datasets.

1 Introduction

Many information extraction tasks – including named entity recognition, relation extraction, event extraction, and coreference resolution – can benefit from incorporating global context across sentences or from non-local dependencies among phrases. For example, knowledge of a coreference relationship can provide information to help infer the type of a difficult-to-classify entity mention. In event extraction, knowledge of the entities present in a sentence can provide information that is useful for predicting event triggers.

To model global context, previous works have used pipelines to extract syntactic, discourse, and other hand-engineered features as inputs to structured prediction models (Li et al., 2013; Yang and Mitchell, 2016; Li and Ji, 2014) and neural scoring functions (Nguyen and Nguyen, 2019), or as a guide for the construction of neural architectures (Peng et al., 2017; Zhang et al., 2018; Sha et al., 2018; Christopoulou et al., 2018). Recent end-to-end systems have achieved strong performance by dynmically constructing graphs of spans whose edges correspond to task-specific relations (Luan et al., 2019; Lee et al., 2018; Qian et al., 2018).

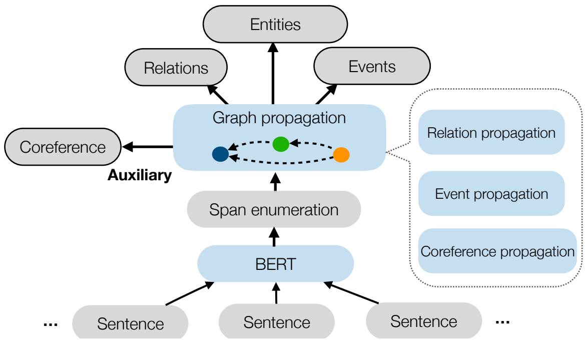


Figure 1: **Overview of our framework: DYGIE++.** Shared span representations are constructed by refining contextualized word embeddings via span graph updates, then passed to scoring functions for three IE tasks.

Meanwhile, contextual language models (Dai and Le, 2015; Peters et al., 2017, 2018; Devlin et al., 2018) have proven successful on a range of natural language processing tasks (Bowman et al., 2015; Sang and De Meulder, 2003; Rajpurkar et al., 2016). Some of these models are also capable of modeling context beyond the sentence boundary. For instance, the attention mechanism in BERT's transformer architecture can capture relationships between tokens in nearby sentences.

In this paper, we study different methods to incorporate global context in a general multi-task IE framework, building upon a previous span-based IE method (Luan et al., 2019). Our DYGIE++ framework, shown in Figure 1, enumerates candidate text

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 5784–5789,
Hong Kong, China, November 3–7, 2019.

spans and encodes them using contextual language models and task-specific message updates passed over a text span graph. Our framework achieves state-of-the results across three IE tasks, leveraging the benefits of both contextualization methods.

We conduct experiments and a thorough analysis of the model on named entity, relation, and event extraction. Our findings are as follows: (1) Our general span-based framework produces state-of-the-art results on all tasks and all but one subtasks across four text domains, with relative error reductions ranging from 0.2 - 27.9%. (2) BERT encodings are able to capture important within and adjacent-sentence context, achieving improved performance by increasing the input window size. (3) Contextual encoding through message passing updates enables the model to incorporate cross-sentence dependencies, improving performance beyond that of BERT alone, especially on IE tasks in specialized domains.

2 Task and Model

Our DyGIE++framework extends a recent span-based model for entity and relation extraction (Luan et al., 2019) as follows: (1) We perform event extraction as an additional task and propagate span updates across a graph connecting event triggers to their arguments. (2) We build span representations on top of multi-sentence BERT encodings.

2.1 Task definitions

The input is a document represented as a sequence of tokens D, from which our model constructs spans $S = \{s_1, \ldots, s_T\}$, the set of all possible within-sentence phrases (up to a threshold length) in the document.

Named Entity Recognition involves predicting the best entity type label e_i for each span s_i. For all tasks, the best label may be a "null" label. Relation Extraction involves predicting the best relation type r_{ij} for all span pairs (s_i, s_j). For the data sets studied in this work, all relations are between spans within the same sentence. The coreference resolution task is to predict the best coreference antecedent c_i for each span s_i. We perform coreference resolution as auxiliary task, to improve the representations available for the "main" three tasks.

Event Extraction involves predicting named entities, event triggers, event arguments, and argument roles. Specifically, each token d_i is predicted as an event trigger by assigning it a label t_i. Then, for each trigger d_i, event arguments are assigned to this event trigger by predicting an argument role a_{ij} for all spans s_j in the same sentence as d_i. Unlike most work on event extraction, we consider the realistic setting where gold entity labels are not available. Instead, we use predicted entity mentions as argument candidates.

2.2 DyGIE++ Architecture

Figure 1 depicts the four-stage architecture. For more details, see (Luan et al., 2019).

Token encoding: DyGIE++ uses BERT for token representations using a "sliding window" approach, feeding each sentence to BERT together with a size-L neighborhood of surrounding sentences.

Span enumeration: Spans of text are enumerated and constructed by concatenating the tokens representing their left and right endpoints, together with a learned span width embedding.

Span graph propagation: A graph structure is generated dynamically based on the model's current best guess at the relations present among the spans in the document. Each span representation $\mathbf{g}_j^t$ is updated by integrating span representations from its neighbors in the graph according to three variants of graph propagation. In coreference propagation, a span's neighbors in the graph are its likely coreference antecedents. In relation propagation, neighbors are related entities within a sentence. In event propagation, there are event trigger nodes and event argument nodes; trigger nodes pass messages to their likely arguments, and arguments pass messages back to their probable triggers. The whole procedure is trained end-to-end, with the model learning simultaneously how to identify important links between spans and how to share information between those spans.

More formally, at each iteration t the model generates an update $\mathbf{u}_x^t(i)$ for span $s^t \in \mathbb{R}^d$:

$$\mathbf{u}_x^t(i) = \sum_{j \in B_x(i)} V_x^t(i,j) \odot \mathbf{g}_j^t, \qquad (1)$$

where $\odot$ denotes elementwise multiplication and $V_x^t(i,j)$ is a measure of similarity between spans i and j under task x – for instance, a score indicating the model's confidence that span j is the coreference antecedent of span i. For relation extraction, we use a ReLU activation to enforce sparsity. The final updated span representation $\mathbf{g}_j^{t+1}$ is computed as a convex combination of the previous representation and the current update, with weights determined by a gating function.

Multi-task classification: The re-contextualized representations are input to scoring functions which make predictions for each of the end tasks. We use a two-layer feedforward neural net (FFNN) as the scoring function. For trigger and named entity prediction for span $\mathbf{g}_i$, we compute $\text{FFNN}_{\text{task}}(\mathbf{g}_i)$. For relation and argument role prediction, we concatenate the relevant pair of embeddings and compute $\text{FFNN}_{\text{task}}([\mathbf{g}_i, \mathbf{g}_j])$.

3 Experimental Setup

Data We experiment on four different datasets: ACE05, SciERC, GENIA and WLPC (Statistics and details on all data sets and splits can be found in Appendix A.). The **ACE05** corpus provides entity, relation, and event annotations for a collection of documents from a variety of domains such as newswire and online forums. For named entity and relation extraction we follow the train / dev / test split from Miwa and Bansal (2016). Since the ACE data set lacks coreference annotations, we train on the coreference annotations from the OntoNotes dataset (Pradhan et al., 2012). For event extraction we use the split described in Yang and Mitchell (2016); Zhang et al. (2019). We refer to this split as ACE05-E in what follows. The **SciERC** corpus (Luan et al., 2018) provides entity, coreference and relation annotations from 500 AI paper abstracts. The **GENIA** corpus (Kim et al., 2003) provides entity tags and coreferences for 1999 abstracts from the biomedical research literature with a substantial portion of entities (24%) overlapping some other entity. The **WLPC** dataset provides entity, relation, and event annotations for 622 wet lab protocols (Kulkarni et al., 2018). Rather than treating event extraction as a separate task, the authors annotate event triggers as an entity type, and event arguments as relations between an event trigger and an argument.

Evaluation We follow the experimental setups of the respective state-of-the-art methods for each dataset: Luan et al. (2019) for entities and relations, and Zhang et al. (2019) for event extraction. An entity prediction is correct if its label and span matches with a gold entity; a relation is correct if both the span pairs and relation labels match with a gold relation triple. An event trigger is correctly identified if its offsets match a gold trigger. An argument is correctly identified if its offsets and event type match a gold argument. Triggers and arguments are correctly classified if their event types and event roles are also correct, respectively.

Dataset	Task	SOTA	Ours	Δ%
ACE05	Entity	88.4	**88.6**	1.7
	Relation	63.2	**63.4**	0.5
ACE05-Event*	Entity	87.1	**90.7**	27.9
	Trig-ID	73.9	**76.5**	9.6
	Trig-C	72.0	**73.6**	5.7
	Arg-ID	**57.2**	55.4	-4.2
	Arg-C	52.4	**52.5**	0.2
SciERC	Entity	65.2	**67.5**	6.6
	Relation	41.6	**48.4**	11.6
GENIA	Entity	76.2	**77.9**	7.1
WLPC	Entity	79.5	**79.7**	1.0
	Relation	64.1	**65.9**	5.0

Table 1: **DYGIE++ achieves state-of-the-art results**. Test set F1 scores of best model, on all tasks and datasets. We define the following notations for events: *Trig*: Trigger, *Arg*: argument, *ID*: Identification, *C*: Classification. * indicates the use of a 4-model ensemble for trigger detection. See Appendix E for details. The results of the single model are reported in Table 2 (c). We ran significance tests on a subset of results in Appendix D. All were statistically significant except Arg-C and Arg-ID on ACE05-Event.

Model Variations We perform experiments with the following variants of our model architecture. **BERT + LSTM** feeds pretrained BERT embeddings to a bi-directional LSTM layer, and the LSTM parameters are trained together with task specific layers. **BERT Finetune** uses supervised fine-tuning of BERT on the end-task. For each variation, we study the effect of integrating different task-specific message propagation approaches.

Comparisons For entity and relation extraction, we compare DYGIE++ against the DYGIE system it extends. DYGIE is a system based on ELMo (Peters et al., 2018) that uses dynamic span graphs to propagate global context. For event extraction, we compare against the method of Zhang et al. (2019), which is also an ELMo-based approach that relies on inverse reinforcement learning to focus the model on more difficult-to-detect events.

Implementation Details Our model is implemented using AllenNLP (Gardner et al., 2017). We use $\text{BERT}_{\text{BASE}}$ for entity and relation extraction tasks and use $\text{BERT}_{\text{LARGE}}$ for event extraction. For BERT finetuning, we use BertAdam with the learning rates of 1×10^{-3} for the task specific layers, and 5.0×10^{-5} for BERT. We use a longer warmup period for BERT than the warmup period for task specific-layers and perform linear decay of the learning rate following the warmup

	ACE05	SciERC	GENIA	WLPC
BERT + LSTM	85.8	69.9	78.4	**78.9**
`+RelProp`	85.7	70.5	-	78.7
`+CorefProp`	86.3	**72.0**	78.3	-
BERT Finetune	87.3	70.5	78.3	78.5
`+RelProp`	86.7	71.1	-	78.8
`+CorefProp`	**87.5**	71.1	**79.5**	-

Table 2: F1 scores on NER.

	ACE05	SciERC	WLPC
BERT + LSTM	60.6	40.3	65.1
`+RelProp`	61.9	41.1	65.3
`+CorefProp`	59.7	42.6	-
BERT FineTune	**62.1**	44.3	65.4
`+RelProp`	62.0	43.0	**65.5**
`+CorefProp`	60.0	**45.3**	-

Table 3: F1 scores on Relation.

	Entity	Trig-C	Arg-ID	Arg-C
BERT + LSTM	90.5	68.9	**54.1**	**51.4**
`+EventProp`	**91.0**	68.4	52.5	50.3
BERT FineTune	89.7	**69.7**	53.0	48.8
`+EventProp`	88.7	68.2	50.4	47.2

Table 4: F1 scores on ACE05-E.

Table 5: **Comparison of contextualization methods**. All ablations are performed on the dev set except for ACE05-E, where the precedent in the literature is to ablate on test.

period. Each of the feed-forward neural networks has two hidden layers and ReLU activations and 0.4 dropout. We use 600 hidden units for event extraction and 150 for entity and relation extraction (more details in Appendix E).

4 Results and Analyses

State-of-the-art Results Table 1 shows test set F1 on the entity, relation and event extraction tasks. Our framework establishes a new state-of-the-art on all three high-level tasks, and on all subtasks except event argument identification. Relative error reductions range from 0.2 - 27.9% over previous state of the art models.

Benefits of Graph Propagation Table 2 shows that Coreference propagation (`CorefProp`) improves named entity recognition performance across all three domains. The largest gains are on the computer science research abstracts of SciERC, which make frequent use of long-range coreferences, acronyms and abbreviations. `CorefProp` also improves relation extraction on SciERC.

Relation propagation (`RelProp`) improves relation extraction performance over pretrained BERT, but does not improve fine-tuned BERT. We believe

Task	Variation	1	3
Relation	BERT+LSTM	59.3	**60.6**
	BERT Finetune	62.0	**62.1**
Entity	BERT+LSTM	90.0	**90.5**
	BERT Finetune	88.8	**89.7**
Trigger	BERT+LSTM	**69.4**	68.9
	BERT Finetune	68.3	**69.7**
Arg Class	BERT+LSTM	48.6	**51.4**
	BERT Finetune	**50.0**	48.8

Table 6: **Effect of BERT cross-sentence context**. F1 score of relation F1 on ACE05 dev set and entity, arg, trigger extraction F1 on ACE05-E test set, as a function of the BERT context window size.

this occurs because all relations are within a single sentence, and thus BERT can be trained to model these relationships well.

Our best event extraction results did not use any propagation techniques (Table 4). We hypothesize that event propagation is not helpful due to the asymmetry of the relationship between triggers and arguments. Methods to model higher-order interactions among event arguments and triggers represent an interesting direction for future work.

Benefits of Cross-Sentence Context with BERT Table 6 shows that both variations of our BERT model benefit from wider context windows. Our model achieves the best performance with a 3-sentence window across all relation and event extraction tasks.

Pre-training or Fine Tuning BERT Under Limited Resources Table 5 shows that fine-tuning BERT generally performs slightly better than using the pre-trained BERT embeddings combined with a final LSTM layer.[1] Named entity recognition improves by an average of 0.32 F1 across the four datasets tested, and relation extraction improves by an average of 1.0 F1, driven mainly by the performance gains on SciERC. On event extraction, fine-tuning decreases performance by 1.6 F1 on average across tasks. We believe that this is due to the high sensitivity of both BERT finetuning and event extraction to the choice of optimization hyperparameters – in particular, the trigger detector begins overfitting before the argument detector is finished training.

Pretrained BERT combined with an LSTM layer and graph propagation stores gradients on 15 million parameters, as compared to the 100 million pa-

[1]Pre-trained BERT without a final LSTM layer performed substantially worse than either fine-tuning BERT, or using pre-trained BERT with a final LSTM layer.

	SciERC		GENIA
	Entity	Relation	Entity
Best BERT	69.8	41.9	78.4
Best SciBERT	**72.0**	**45.3**	**79.5**

Table 7: In-domain pre-training: SciBERT vs. BERT

rameters in $\text{BERT}_{\text{BASE}}$. Since the BERT + LSTM + Propagation approach requires less memory and is less sensitive to the choice of optimization hyperparameters, it may be appealing for non-experts or for researchers working to quickly establish a reasonable baseline under limited resources. It may also be desirable in situations where fine-tuning BERT would be prohibitively slow or memory-intensive, for instance when encoding long documents like scientific articles.

Importance of In-Domain Pretraining We replaced BERT (Devlin et al., 2018) with SciBERT (Beltagy et al., 2019) which is pretrained on a large multi-domain corpus of scientific publications. Table 7 compares the results of BERT and SciBERT with the best-performing model configurations. SciBERT significantly boosts performance for scientific datasets including SciERC and GENIA. These results indicate that introducing unlabeled text of similar domains for pre-training can significantly improve performance.

Qualitative Analysis To better understand the mechanism by which graph propagation improved performance, we examined all named entities in the SciERC dev set where the prediction made by the BERT + LSTM + `CorefProp` model from Table 2 was different from the BERT + LSTM model. We found 44 cases where the `CorefProp` model corrected an error made by the base model, and 21 cases where it introduced an error. The model without `CorefProp` was often overly specific in the label it assigned, labeling entities as *Material* or *Method* when it should have given the more general label *Other Scientific Term*. Visualizations of the disagreements between the two model variants can be found in Appendix C. Figure 2 shows an example where span updates passed along a coreference chain corrected an overly-specific entity identification for the acronym "CCRs". We observed similar context sharing via `CorefProp` in the GENIA data set, and include an example in Appendix C.

Coreference propagation updated the span representations of all but one of 44 entities, and in 68% of these cases the update with the largest coreference "attention weight" came from a text span in a different sentence that was itself a named entity.

1: This paper summarizes the formalism of **Category Cooccurrence Restrictions (CCRs)** and describes two parsing algorithms that interpret it .

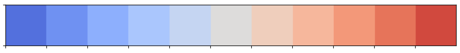
2: **CCRs** are Boolean conditions on the cooccurrence of categories in local trees which allow the statement of generalizations which can not be captured in other current syntax formalisms .

(a) The green span *CCRs* in sentence 2 is updated based on its predicted coreference antecedent.

2: **CCRs** are Boolean conditions on the cooccurrence of categories in local trees which allow the statement of generalizations which can not be captured in other current syntax formalisms .

3: The use of **CCRs** leads to syntactic descriptions formulated entirely with restrictive statements .

(b) The mention of *CCRs* in sentence 2 serves as a bridge to propagate information from sentence 1 to the mention of *CCRs* in sentence 3



(c) Coreference link strength. Red is strong.

Figure 2: **`CorefProp` enables a correct entity prediction.** In each subplot, the green token is being updated by coreference propagation. The preceeding tokens are colored according to the strength of their predicted coreference links with the green token. Tokens in **bold** are part of a gold coreference cluster discussing *CCRs*. During the `CorefProp` updates, the span *CCRs* in sentence 2 is updated based on its antecedent *Category Cooccurrence Restrictions*. Then, it passes this information along to the span *CCRs* in sentence 3. As a result, the model changes its prediction for *CCRs* in sentence 3 from *Method* – which is overly specific according to the SciERC annotation guideline – to the correct answer *Other Scientific Term*.

5 Conclusion

In this paper, we provide an effective plug-and-play IE framework that can be applied to many information extraction tasks. We explore the abilities of BERT embeddings and graph propagation to capture context relevant for these tasks. We find that combining these two approaches improves performance compared to using either one alone, with BERT building robust multi-sentence representations and graph propagations imposing additional structure relevant to the problem and domain under consideration. Future work could extend the framework to other NLP tasks and explore other approaches to model higher-order interactions like those present in event extraction.

Acknowledgments

This research was supported by the ONR MURI N00014-18-1-2670, ONR N00014-18-1-2826, DARPA N66001-19-2-4031, NSF (IIS 1616112), Allen Distinguished Investigator Award, and Samsung GRO. We thank Mandar Joshi for his valuable BERT finetuning advice, Tongtao Zhang for sharing the ACE data code, anonymous reviewers, and the UW-NLP group for their helpful comments.

References

Iz Beltagy, Arman Cohan, and Kyle Lo. 2019. Scibert: Pretrained contextualized embeddings for scientific text. *ArXiv*, abs/1903.10676.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2018. A walk-based model on entity graphs for relation extraction. In *ACL*.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *NeurIPs*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. Genia corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19 Suppl 1:i180–2.

Chaitanya Kulkarni, Wei Xu, Alan Ritter, and Raghu Machiraju. 2018. An annotated corpus for machine reading of instructions in wet lab protocols. In *NAACL-HLT*.

Kenton Lee, Luheng He, and Luke S. Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *NAACL-HLT*.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *ACL*.

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

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *EMNLP*.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *NAACL-HLT*.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *ACL*.

Trung Minh Nguyen and Thien Huu Nguyen. 2019. One for all: Neural joint modeling of entities and events. In *AAAI*.

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

Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *ACL*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 1–40. Association for Computational Linguistics.

Yujie Qian, Enrico Santus, Zhijing Jin, Jiang Guo, and Regina Barzilay. 2018. Graphie: A graph-based framework for information extraction. In *NAACL-HLT*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *NAACL*.

Lei Sha, Feng Qian, Baobao Chang, and Zhifang Sui. 2018. Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction. In *AAAI*.

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *HLT-NAACL*.

Tongtao Zhang, Heng Ji, and Avirup Sil. 2019. Joint entity and event extraction with generative adversarial imitation learning. *Data Intelligence*, 1:99–120.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *EMNLP*.