A Two-Step Approach for Implicit Event Argument Detection

Zhisong Zhang, Xiang Kong, Zhengzhong Liu, Xuezhe Ma, Eduard Hovy
Language Technologies Institute, Carnegie Mellon University
{zhisongz,xiangk,liu,xuezhem,hovy}@cs.cmu.edu

Abstract

In this work, we explore the implicit event argument detection task, which studies event arguments beyond sentence boundaries. The addition of cross-sentence argument candidates imposes great challenges for modeling. To reduce the number of candidates, we adopt a two-step approach, decomposing the problem into two sub-problems: argument head-word detection and head-to-span expansion. Evaluated on the recent *RAMS* dataset (Ebner et al., 2020), our model achieves overall better performance than a strong sequence labeling baseline. We further provide detailed error analysis, presenting where the model mainly makes errors and indicating directions for future improvements. It remains a challenge to detect implicit arguments, calling for more future work of document-level modeling for this task.

1 Introduction

Event argument detection is a key component in the task of event extraction. It resembles semantic role labeling (SRL) in that the main target is to find argument spans to fill the roles of event frames. However, event arguments can go beyond sentence boundaries: there can be *non-local* or *implicit* arguments at the document level. Figure 1 shows such an example: for the *purchase* event, which is triggered by the word "bought", its *money* argument appears in the previous sentence.

Implicit arguments have been under-explored in event extraction. Most of previous systems (Li et al., 2013; Chen et al., 2015; Nguyen et al., 2016; Wang et al., 2019) only consider local arguments in the same sentence of the event trigger. While incorporating implicit arguments requires corresponding annotations, few exists in most of the widely used event datasets, like ACE2005 (LDC, 2005; Walker et al., 2006) and RichERE (LDC, 2015). There are several annotation efforts for implicit arguments in SRL, including *G&C* (Gerber and Chai, 2010, 2012), *SemEval-2010* (Ruppenhofer et al., 2009, 2010), and *80Days* (Feizabadi and Padó, 2014). Yet most are performed with different ontologies such as Nombank (*G&C*) and FrameNet (*SemEval-2010* and *80Days*); on different domains (e.g. novels); and in smaller scales (*G&C* and *80Days* only cover 10 types of predicates). The lack of annotations poses challenges to train and transfer implicit argument models for event extraction.

Recently, Ebner et al. (2020) create the Roles Across Multiple Sentences (*RAMS*) dataset, which covers multi-sentence implicit arguments for a wide range of event and role types. They further develop a span-based argument linking model and achieve relatively high scores. However, they mainly explore a simplified setting that assumes

(a)	The new computer cost 3000 dollars, while the old one cost 1000 dollars. Nevertheless, he still **bought** the more expensive one.
(b)	The new computer cost 3000 dollars, while the old one cost 1000 dollars. Therefore, he **bought** the cheaper one.

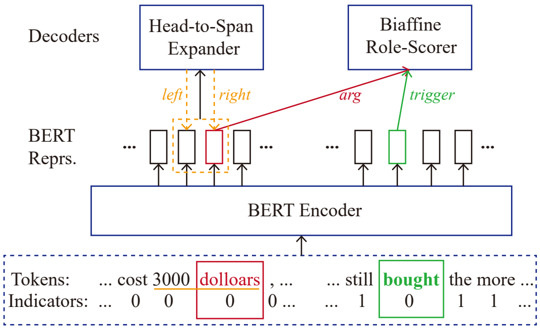
Decoders
Head-to-Span Expander
Biaffine Role-Scorer
left
right
arg
trigger
BERT Reprs.
BERT Encoder
Tokens: ... cost 3000 dolloars , ... still bought the more ...
Indicators: ... 0 0 0 0 ... ... 1 0 1 1 ...

Figure 1: Examples of implicit arguments and model illustration. The **bold** text indicates the trigger word for the *purchase* event, while the underlined text indicates its non-local "*money*" argument in the previous sentence. Our model first detects the head-word "dollars", and then expands it to the whole span.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7479–7485
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

the availability of gold argument spans. We extend their work and explore the more challenging full detection problem that predicts argument spans among all possible candidates. The difficulty of the full problem is highlighted in Figure 1. Both "3000 dollars" and "1000 dollars" are good candidates for the *money* role of the *purchase* event, but the selections are different given different contexts.

When considering all possible candidate spans that may occur in any sentences, their quadratic number poses great challenges for the detection. Inspired by dependency-based SRL (Surdeanu et al., 2008; Hajič et al., 2009), we take the syntactical head-words as the proxy for full argument spans, hypothesizing that the head-words can contain enough information to fill the argument roles. Based on this, we adopt a two-step approach: first detecting the head-words of the arguments, and adopting a second step of head-to-span expansion. Actually, this type of two-step setup is not uncommon in prior work of information extraction, including entity detection (Lin et al., 2019), coreference resolution (Peng et al., 2015) and document-level pseudo-coreference (Jauhar et al., 2015; Liu et al., 2016). By considering only individual tokens in the detection step, the system only needs to handle a candidate space whose size scales linearly in respective to the number of tokens instead of quadratically.

With the same setting of fine-tuning BERT (Devlin et al., 2019) encoder, we show the effectiveness of our model by obtaining overall better results than a strong sequence-labeling model. We further provide detailed error analysis, showing that the main difficulties of the task are upon non-local and non-core arguments. Our analysis shows that the implicit argument task is quite challenging, calling for more future work on document-level semantic understanding for this task.

2 Model

The goal of event argument detection is to create labeled links between argument spans and the predicate (event trigger). Recent state-of-the-art solutions for sentence-level SRL perform the detection in an end-to-end setting, such as span-based (He et al., 2018; Ouchi et al., 2018), and sequence labeling models (He et al., 2017; Shi and Lin, 2019). However, span-based models face great challenges when considering arguments across sentence boundaries, since the computational complexity of such models grows quadratically to deal with $O(N^2)$ span candidates given N tokens. While traditional sequence labeling models can run in linear-time, they are less flexible and extensible in complex scenarios like overlapping mentions and multiple roles for one mention. In this work, we take a two-step approach that decomposes the problem explicitly into two sub-problems, based on the hypothesis that head-words can usually capture the information of the mention spans. Figure 1 illustrates the three main modules of our model: 1) BERT-based Encoder, 2) Argument Head-Word Detector, and 3) Head-to-span Expander.

2.1 BERT-based Encoder

Our encoding module is a BERT-based contextualized encoder. The input contains a predicate word (or occasionally a span), which triggers an event, together with its multi-sentence context. We refer to the sentence containing the event trigger as the *center sentence*. We concatenate the tokens within the 5-sentence window (the window size used in *RAMS* annotation) of the center sentences, and feed them to BERT to obtain the contextual representation $\mathbf{e}$ of each token. In addition, we add special `token_type_ids` indicators: tokens of the event trigger are assigned 0, other tokens in the center sentence get 1, and tokens in surrounding sentences get 0[1]. We only adopt the indicators when fine-tuning BERT, since the pre-trained BERT originally uses them as segment ids.

2.2 Argument Head-word Detector

Instead of directly deciding argument spans, we first identify the head-words of the arguments. The hypothesis is that the head-word is able to represent the meaning of the whole span. In this way, this sub-problem mimics a token-pairwise dependency-parsing problem. Following (Dozat and Manning, 2017, 2018), we adopt a biaffine module to calculate $\Pr_r(p, c)$: the probability of a candidate word c filling an argument role r in the frame governed by a predicate p. We first take the contextualized representations of the candidate ($\mathbf{e}_c$) and the predicate ($\mathbf{e}_p$), which are calculated by BERT as described in §2.1. "Biaffine$_r$" further gives the pairwise score based on these representations, and $\Pr_r(p, c)$ is then

[1] We overload 0 because pre-trained BERT only has two types of `token_type_id`. Nevertheless, the trigger words are still distinguishable since they appear inside center sentences, and are separated from other sentences.

given by softmax with the scores:

$$\Pr_r(p,c) = \frac{\exp \mathrm{Biaffine}_r(\mathbf{e}_p, \mathbf{e}_c)}{\sum_{c' \in \mathcal{C} \cup \{\epsilon\}} \exp \mathrm{Biaffine}_r(\mathbf{e}_p, \mathbf{e}_{c'})}$$

where the normalization is done over the argument candidate set $\mathcal{C}$ (or null ϵ, whose score is fixed to 0) for each role, following (Ebner et al., 2020; Ouchi et al., 2018). During training, we use the cross-entropy loss to guide the network to pick head-words of gold arguments (or ϵ if there are no arguments for this role). If there are multiple arguments for one role, we view them as individual instances and sum the losses. At inference time, we simply pick the maximumly-scored argument (or ϵ) for each role.

2.3 Head-to-span Expander

The second module expands each head-word of the argument to its full span. We view it as a combination of left and right boundary classification problems. Taking the left-expanding scenario (L) as example, for each head-word h, we generate a set of candidate spans by adding words one by one on the left up to K words (we empirically set $K = 7$), and calculate the probability of word b being the boundary as follow:

$$\Pr_L(h,b) = \frac{\exp \mathrm{MLP}_L(\mathbf{e}_h, \mathbf{e}_b)}{\sum_{b' \in (h-K,h]} \exp \mathrm{MLP}_L(\mathbf{e}_h, \mathbf{e}_{b'})}$$

Here, the input to the Multi-layer Perceptron (MLP) is again the contextualized representations as depicted in §2.1. During training, we minimize cross-entropy losses on the left and right respectively. At test time, we expand to the maximumly-scored boundary words on both sides.

3 Experiment

We conduct all experiments[2] on the *RAMS* (v1.0) dataset and focus on the event argument detection task: given (gold) event triggers and their multi-sentence contexts, predicting the argument spans from raw input tokens. Following (Ebner et al., 2020), we only use gold event types in the type-constrained decoding (TCD) setting.

Through our experiments, we adopt the pre-trained `bert-base-cased` model. We train all the models for maximumly 20 epochs. If fine-tuning BERT, we set the initial learning rate to 5e-5; otherwise, it is set to 2e-4. We jointly train our argument-detector and span-expander, with loss multipliers of 1.0 and 0.5, respectively.

	+TCD	Dev. F1	Test P	Test R	Test F1
Span	no	69.9	62.8	74.9	68.3
	yes	75.1	78.1	69.2	73.3
Head	no	71.0	71.5	66.2	68.8
	yes	74.3	81.1	66.2	73.0

Table 1: Comparison of Span-based (Ebner et al., 2020) and Head-based (ours) models on *RAMS*, given gold argument spans. "+TCD" indicates whether applying type-constrained decoding based on gold event types.

Since head-words are not annotated, we apply a simple rule: utilizing predicted dependency trees, we heuristically pick the word that has the smallest arc distance to the dependency root as the head. Ties are broken by choosing the rightmost one. There are cases where this procedure does not always give the perfect head, or there is no single head-word for a span (e.g., in multi-word expressions or conjunction). Nevertheless, we find this strategy works well in practice.

3.1 Argument Linking with Gold Spans

Setting To compare our model with span-based models, we first evaluate in the same setting of (Ebner et al., 2020) that assumes gold argument spans. We directly apply the head rule on the gold spans and consider the head-words as candidates. We also adopt the same BERT setting: learning a linear combination of layers 9, 10, 11 and 12, and applying neither the special input indicators nor fine-tuning.

Results Table 1 compares our results with the reported results of the span-based model from (Ebner et al., 2020). The results show that the head-word approach can get comparable results to the span-based counterpart. This matches our hypothesis that head-words contain sufficient information of surrounding words using contextualized embedding, making them reasonable alternatives to full argument spans.

3.2 Full Argument Detection

Setting This setting considers all arguments from any spans in the multi-sentence context. Unless otherwise noted, here we use the last layer of BERT and apply fine-tuning for the whole model. We compare with a strong BERT-based BIO-styled sequence labeling model (Shi and Lin, 2019). We

[2] Our implementation is publicly available at `https://github.com/zzsfornlp/zmsp`

	+TCD	Dev.		Test	
		SpanF1	HeadF1	SpanF1	HeadF1
Seq.	no	$38.1_{\pm 0.7}$	$45.7_{\pm 0.7}$	$39.3_{\pm 0.4}$	$47.1_{\pm 0.7}$
	yes	$39.2_{\pm 0.7}$	$46.7_{\pm 0.8}$	$40.5_{\pm 0.4}$	$48.0_{\pm 0.5}$
Head	no	$38.9_{\pm 0.6}$	$46.4_{\pm 0.7}$	$40.1_{\pm 0.7}$	$47.7_{\pm 0.9}$
	yes	$40.3^{*}_{\pm 0.6}$	$48.0^{*}_{\pm 0.7}$	$41.8^{*}_{\pm 0.6}$	$49.7^{*}_{\pm 0.8}$

Table 2: Comparison of the sequence-labeling model (Seq.) and our Head-based model for argument detection on *RAMS* v1.0. All results are averaged over five runs, '$*$' denotes that the result of Head model is significantly better than the corresponding Seq. model (by paired randomization test, $p < 0.05$).

	SpanF1	HeadF1
BERT-Full	$38.9_{\pm 0.6}$	$46.4_{\pm 0.7}$
No-Indicator	$35.6_{\pm 0.4}$	$42.9_{\pm 0.4}$
No-FineTuning	$34.4_{\pm 0.5}$	$40.0_{\pm 0.4}$
LSTM	$26.6_{\pm 0.4}$	$31.9_{\pm 0.6}$

Table 3: Ablation on the encoder for the head-based argument detection model (on development set, no type-constrained decoding). "BERT-Full" is our full fine-tuned BERT encoder, "No-Indicator" ablates indicating inputs, "No-FineTuning" freezes all pre-trained parameters of BERT, and "LSTM" replaces the BERT with a bi-directional LSTM encoder.

adopt a modified version[3] from AllenNLP and retrain it on *RAMS* with similar settings: adopting special input indicators and fine-tuning BERT. For arguments that have multiple roles labels, we simply concatenate the labels as a new class.

Results Table 2 shows the main results for full argument detection. Since the criterion of full-span matching might be too strict in some way, we also report head-word based F1 scores by evaluating solely on head-word matches (obtained using the same head rules). The results show that our head-word based approach gets better results on average without type-constrained decoding and significantly better results after adopting type-constrained decoding with gold event types. Our head-driven approach is also flexible and easily extensible to more complex scenarios like nesting mentions or multiple roles, while keeping the linear complexity.

Ablation Table 3 lists the ablation results on the encoder. The results show that the BERT encoder contributes much to the performance of our full model. Fine-tuning BERT and the special indicator inputs can provide further improvements.

	d=-2 (3.6%)	d=-1 (7.5%)	d=0 (82.8%)	d=1 (4.0%)	d=2 (2.1%)
Seq.	$14.0_{\pm 0.6}$	$14.0_{\pm 2.4}$	$41.2_{\pm 0.9}$	$15.7_{\pm 1.0}$	$4.2_{\pm 2.5}$
Head	$15.6_{\pm 1.7}$	$15.3_{\pm 1.0}$	$43.4_{\pm 0.7}$	$17.8_{\pm 2.6}$	$8.5_{\pm 6.2}$

Table 4: Performance breakdown for Span-F1 by argument-trigger distance d (on development set, no type-constrained decoding). Numbers in parentheses at the second row indicate the distribution over distance d.

On Sentence Distances Table 4 lists the performance breakdown on different sentence distances between arguments and triggers. As opposed to the relative consistent performance in the gold span setting, as shown in (Ebner et al., 2020), we notice a dramatic performance drop on non-local arguments. There may be two main reasons: 1) data imbalance, since non-local implicit arguments appear much less frequently (only around 18% in *RAMS*) than local ones; 2) lack of direct syntax signals, making the connections between the implicit arguments and event triggers much weaker than the local ones.

On Argument Roles We also investigate performance breakdowns on different argument roles. The results are shown in Figure 2, where we take the top-20 frequent roles to get more robust results. We can observe that our model performs better on core roles such as "*communicator*", "*employee*" and "*victim*" (with F1 > 50), but struggles on non-core roles, like "*instrument*", "*origin*" and "*destination*", with F1 scores of around 20 to 30. The F1 scores correlate well (with Pearson and Spearman correlation coefficients of 0.64 and 0.70, respectively) with the local percentages: the more often one role appears locally around the event trigger, the better results it can obtain. These patterns are not surprising if we consider the possible underlying reasoning. The non-core arguments are not closely related with the event trigger, and thus can appear more freely at other places (or sometimes even be omitted), leading to a lower local percentage and also being harder to detect.

3.3 Manual Analysis

To further investigate in detail what type of errors the model makes, we sample 200 event frames from the development set and manually compare our model's predictions with the gold annotations. Overall, there are 459 annotated arguments and 442

[3] `https://github.com/allenai/allennlp/blob/b89ff098372656b674ec71457dda071222fd05ae/allennlp/models/srl_bert.py`

Category	Description	Example	Count (Percentage)
Correct	Correct	-	348 (38.6%)
Span	Unimportant span mismatch	The [monument]$_{\text{artifact}}$ to fallen Soviet sailors$_{\text{artifact}}$ in Limbazi, was **demolished**$_{\text{Destroy}}$ by activists.	82 (9.1%)
Coref.	Co-references	The United States$_{\text{destination}}$ gets more energy domestically, as [the country]$_{\text{destination}}$ continues to rely on oil **imports**$_{\text{Tranport}}$ from elsewhere.	60 (6.7%)
Possi.	Possible annotation problems	A Chinese official$_{\text{participant}}$ said **dialogue**$_{\text{Discussion}}$ was needed to resolve issues on the Korean peninsula.	44 (4.9%)
Partial	Partially correct	[His]$_{\text{recipient}}$ family, advisers and allies$_{\text{recipient}}$ set about **acquiring**$_{\text{Purchase}}$ expensive overseas homes and positions in the country.	26 (2.9%)
Frame	Frame errors	Relation was wrecked last November when [Turkey]$_{\text{killer attacker}}$ **shot**$_{\text{LifeDie}}$ down a fighter jet over the boarder.	31 (3.4%)
Others	Other errors	-	310 (34.4%)

Table 5: Examples and results of error analysis. In the examples, the **bold** text indicates the trigger word, followed by its event type noted in green. Arguments in gold annotations are indicated by the underlined spans with red role types, while the predicted arguments are indicated by [bracketed] spans with blue role types.

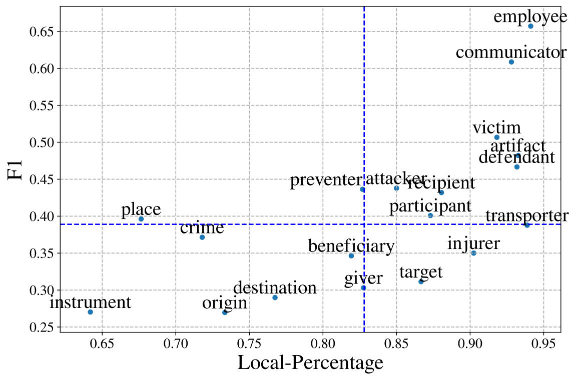

Figure 2: Performance breakdown of Span-F1 on the top-20 frequent roles (on development set, no type-constrained decoding). x-axis represents the percentage of local arguments for this role, while y-axis denotes the role specific Span-F1 scores. The two blue dashed lines denote the overall F1 scores (0.389) and local percentage (82.8%).

predicted ones. For both annotated and predicted arguments, we assign them to one of seven categories, and the results are listed in Table 5. Here, the "Span" errors denote unimportant span mismatches, and they take nearly 9% of all items. If we ignore these errors, the performance can reach around 47%, which roughly matches the automatically evaluated Head-F1 scores. In some way, this supports our intuition to adopt a two-step approach, since the decisions of the span ranges may be separated from the core problem of argument detection, where head-words can be reasonable representatives. Another major source of errors comes from "Coref.", which is not surprising since the same entities can have multiple appearances at the document level. Our analysis indicates that this is a problem that should be further investigated for both modeling and evaluation. Another notable type of error is frame mismatch ("Frame"). In the main setting (without type-constrained decoding), our model neither utilizes nor predicts event frame types, meaning that the frame information purely comes from the trigger words. Therefore, roles belonging to other event frames may be predicted. Finally, the "Others" category includes the ones where we cannot find obviously intuitive patterns. We would identify most of them as the more difficult cases, whose error breakdown follows similar patterns to the overall ones as shown in Figure 2.

4 Conclusion

In this work, we propose a flexible two-step approach for implicit event argument detection. Our head-word based approach effectively reduces the candidate size and achieves good results on the *RAMS* dataset. We further provide detailed error analysis, showing that non-local and non-core arguments are the main difficulties. We hope that this work can shed some light and inspire future work at this line of research.

Acknowledgment

This research was supported in part by DARPA grant FA8750-18-2-0018 funded under the AIDA program. We thank the three anonymous reviewers for their helpful comments.

References

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176, Beijing, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *ICLR*.

Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Parvin Sadat Feizabadi and Sebastian Padó. 2014. Crowdsourcing annotation of non-local semantic roles. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 226–230, Gothenburg, Sweden. Association for Computational Linguistics.

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

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

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18, Boulder, Colorado. Association for Computational Linguistics.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 364–369, Melbourne, Australia. Association for Computational Linguistics.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483, Vancouver, Canada. Association for Computational Linguistics.

Sujay Kumar Jauhar, Raul Guerra, Edgar Gonzàlez Pellicer, and Marta Recasens. 2015. Resolving discourse-deictic pronouns: A two-stage approach to do it. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 299–308, Denver, Colorado. Association for Computational Linguistics.

LDC. 2005. ACE (automatic content extraction) english annotation guidelines for events version 5.4.3. *Linguistic Data Consortium*.

LDC. 2015. Deft Rich ERE annotation guidelines: Events version 3.0. *Linguistic Data Consortium*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria. Association for Computational Linguistics.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2019. Sequence-to-nuggets: Nested entity mention detection via anchor-region networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5182–5192, Florence, Italy. Association for Computational Linguistics.

Zhengzhong Liu, Edgar Gonzàlez Pellicer, and Daniel Gillick. 2016. Exploring the steps of verb phrase ellipsis. In *Proceedings of the Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2016)*, pages 32–40, San Diego, California. Association for Computational Linguistics.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309, San Diego, California. Association for Computational Linguistics.

Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. 2018. A span selection model for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1630–1642, Brussels, Belgium. Association for Computational Linguistics.

Haoruo Peng, Kai-Wei Chang, and Dan Roth. 2015. A joint framework for coreference resolution and mention head detection. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 12–21, Beijing, China. Association for Computational Linguistics.

Josef Ruppenhofer, Caroline Sporleder, Roser Morante, Collin Baker, and Martha Palmer. 2009. SemEval-2010 task 10: Linking events and their participants in discourse. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pages 106–111, Boulder, Colorado. Association for Computational Linguistics.

Josef Ruppenhofer, Caroline Sporleder, Roser Morante, Collin Baker, and Martha Palmer. 2010. SemEval-2010 task 10: Linking events and their participants in discourse. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 45–50, Uppsala, Sweden. Association for Computational Linguistics.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The CoNLL 2008 shared task on joint parsing of syntactic and semantic dependencies. In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 159–177, Manchester, England. Coling 2008 Organizing Committee.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. ACE 2005 multilingual training corpus. *Linguistic Data Consortium*, 57.

Xiaozhi Wang, Ziqi Wang, Xu Han, Zhiyuan Liu, Juanzi Li, Peng Li, Maosong Sun, Jie Zhou, and Xiang Ren. 2019. HMEAE: Hierarchical modular event argument extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5776–5782, Hong Kong, China. Association for Computational Linguistics.