

Towards Open Domain Event Trigger Identification using Adversarial Domain Adaptation

Aakanksha Naik

Carnegie Mellon University
anaik@cs.cmu.edu

Carolyn Rosé

Carnegie Mellon University
cprose@cs.cmu.edu

Abstract

We tackle the task of building supervised event trigger identification models which can generalize better across domains. Our work leverages the adversarial domain adaptation (ADA) framework to introduce domain-invariance. ADA uses adversarial training to construct representations that *are predictive* for trigger identification, but *not predictive* of the example’s domain. It requires no labeled data from the target domain, making it completely unsupervised. Experiments with two domains (English literature and news) show that ADA leads to an average F1 score improvement of 3.9 on out-of-domain data. Our best performing model (BERT-A) reaches 44-49 F1 across both domains, using **no** labeled target data. Preliminary experiments reveal that finetuning on 1% labeled data, followed by self-training leads to substantial improvement, reaching 51.5 and 67.2 F1 on literature and news respectively.¹

1 Introduction

Events are a key semantic phenomenon in natural language understanding. They embody a basic function of language: the ability to report happenings. Events are a basic building block for narratives across multiple domains such as news articles, stories and scientific abstracts, and are important for many downstream tasks such as question answering (Saurí et al., 2005) and summarization (Daniel et al., 2003). Despite their utility, event extraction remains an onerous task. A major reason for this is that the notion of what counts as an “event” depends heavily on the domain and task at hand. For example, should a system which extracts events from doctor notes only focus on medical events (eg: symptoms, treatments), or also annotate lifestyle events (eg: dietary changes, ex-

ercise habits) which may have bearing on the patient’s illness? To circumvent this, prior work has mainly focused on annotating specific categories of events (Grishman and Sundheim, 1996; Dodington et al., 2004; Kim et al., 2008) or narratives from specific domains (Pustejovsky et al., 2003; Sims et al., 2019). This has an important implication for supervised event extractors: they do not generalize to data from a different domain or containing different event types (Keith et al., 2017). Conversely, event extractors that incorporate syntactic rule-based modules (Saurí et al., 2005; Chambers et al., 2014) tend to overgenerate, labeling most verbs and nouns as events. Achieving a balance between these extremes will help in building generalizable event extractors, a crucial problem since annotated training data may be expensive to obtain for every new domain.

Prior work has explored unsupervised (Huang et al., 2016; Yuan et al., 2018), distantly supervised (Keith et al., 2017; Chen et al., 2017; Araki and Mitamura, 2018; Zeng et al., 2018) and semi-supervised approaches (Liao and Grishman, 2010; Huang and Riloff, 2012; Ferguson et al., 2018), which largely focus on automatically generating in-domain training data. In our work, we try to leverage annotated training data from other domains. Motivated by the hypothesis that events, despite being domain/ task-specific, often occur in similar contextual patterns, we try to inject lexical domain-invariance into supervised models, improving generalization, while not overpredicting events.

Concretely, we focus on event trigger identification, which aims to identify triggers (words) that instantiate an event. For example, in “John was born in Sussex”, *born* is a trigger, invoking a BIRTH event. To introduce domain-invariance, we adopt the adversarial domain adaptation (ADA) framework (Ganin and Lempitsky, 2015) which constructs representations that *are predictive* for trigger

¹Our system is available at <https://github.com/aakanksha19/ODETTE>

identification, but *not predictive* of the example’s domain, using adversarial training. This framework requires no labeled target domain data, making it completely unsupervised. Our experiments with two domains (English literature and news) show that ADA makes supervised models more robust on out-of-domain data, with an average F1 score improvement of 3.9, at no loss of in-domain performance. Our best performing model (BERT-A) reaches 44-49 F1 across both domains using **no** labeled data from the target domain. Further, preliminary experiments demonstrate that finetuning on 1% labeled data, followed by self-training leads to substantial improvement, reaching 51.5 and 67.2 F1 on literature and news respectively.

2 Approaching Open Domain Event Trigger Identification

Throughout this work, we treat the task of event trigger identification as a token-level classification task. For each token in a sequence, we predict whether it is an event trigger. To ensure that our trigger identification model can transfer across domains, we leverage the adversarial domain adaptation (ADA) framework (Ganin and Lempitsky, 2015), which has been used in several NLP tasks (Ganin et al., 2016; Li et al., 2017; Liu et al., 2017; Chen et al., 2018; Shah et al., 2018; Yu et al., 2018).

2.1 Adversarial Domain Adaptation

Figure 1 gives an overview of the ADA framework for event trigger identification. It consists of three components: i) representation learner (R) ii) event classifier (E) and iii) domain predictor (D). The representation learner generates token-level representations, while the event classifier and domain predictor use these representations to identify event triggers and predict the domain to which the sequence belongs. The key idea is to train the representation learner to generate representations which *are predictive* for trigger identification but *not predictive* for domain prediction, making it more domain-invariant. A notable benefit here is that the only data we need from the target domain is unlabeled data.

To ensure domain-invariant representation learning, ADA uses adversarial training. Assume that we have a labeled source domain dataset D^s with examples $\{(x_1^s, e_1^s), \dots, (x_n^s, e_n^s)\}$, where x_i^s is the token sequence and e_i^s is the sequence of event tags. We construct auxiliary dataset D^a with examples

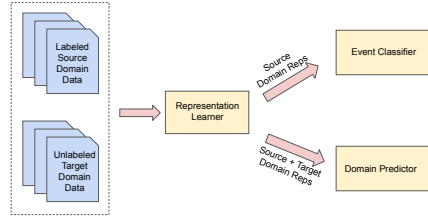


Figure 1: Adversarial Domain Adaptation Framework for Event Trigger Identification

$\{(x_1^a, d_1^a), \dots, (x_n^a, d_n^a)\}$, where x_i^a is the token sequence and d_i^a is the domain label, using token sequences from D^s and unlabeled target domain sentences. The representation learner R maps a token sequence $x_i = (x_{i1}, \dots, x_{ik})$ into token representations $h_i = (h_{i1}, \dots, h_{ik})$. The event classifier E maps representations $h_i = (h_{i1}, \dots, h_{ik})$ to event tags $e_i = (e_{i1}, \dots, e_{ik})$. The domain predictor D creates a pooled representation $p_i = Pool(h_{i1}, \dots, h_{ik})$ and maps it to domain label d_i^a . Given this setup, we apply an alternating optimization procedure. In the first step, we train the domain predictor using D^a , to optimize the following loss:

$$\operatorname{argmin}_D \mathcal{L}(D(h_i^a), d_i^a)$$

In the second step, we train the representation learner and event classifier using D^s to optimize the following loss:

$$\operatorname{argmin}_{R,E} \left[\sum_k \mathcal{L}(E(h_{ik}^s), e_{ik}^s) - \lambda \mathcal{L}(D(h_i^s), d_i^s) \right]$$

\mathcal{L} refers to the cross-entropy loss and λ is a hyperparameter. In practice, the optimization in the above equation is performed using a gradient reversal layer (GRL) (Ganin and Lempitsky, 2015). A GRL works as follows. During the forward pass, it acts as the identity, but during the backward pass it scales the gradients flowing through by $-\lambda$. We apply a GRL g_λ before mapping the pooled representation to a domain label using D . This changes the optimization to:

$$\operatorname{argmin}_{R,E} \left[\mathcal{L}(D(g_\lambda(p_i^s)), d_i^s) + \sum_k \mathcal{L}(E(h_{ik}^s), e_{ik}^s) \right]$$

In our setup, the event classifier and domain predictors are MLP classifiers. For the representation learner, we experiment with several architectures.

2.2 Representation Learner Models

We experiment with the following models:²

LSTM: A unidirectional LSTM over tokens represented using word embeddings.

²Complete implementation details in the appendix

Statistic	LitBank	TimeBank
#Docs	100	183
#Tokens	210,532	80,281
#Events	7849	8103
Event Density	3.73%	10.10%

Table 1: Dataset Statistics

Model	In-Domain			Out-of-Domain		
	P	R	F1	P	R	F1
LSTM	61.9	61.5	61.7	86.1	17.1	28.5
LSTM-A	61.1	61.6	61.3	89.0	18.9	31.2
BiLSTM	64.5	61.7	63.1	91.8	14.4	24.9
BiLSTM-A	66.1	62.8	64.4	92.9	18.5	30.9
POS	74.1	51.9	61.1	93.5	9.6	17.4
POS-A	69.6	57.7	63.1	92.5	15.2	26.1
BERT	73.5	72.7	73.1	88.1	28.2	42.7
BERT-A	71.9	71.3	71.6	85.0	35.0	49.6

Table 2: Model performance on domain transfer experiments from LitBank to TimeBank. Presence of the -A suffix indicates that the model uses adversarial training.

BiLSTM: A bidirectional LSTM over word embeddings to incorporate both left and right context.
POS: A BiLSTM over token representations constructed by concatenating word embeddings with embeddings corresponding to part-of-speech tags. This model explicitly introduces syntax.

BERT: A BiLSTM over contextual token representations extracted using BERT (Devlin et al., 2019), similar to the best-performing model on LitBank, reported by Sims et al. (2019).

3 Experiments

3.1 Datasets

In our experiments, we use the following datasets:³

- **LitBank** (Sims et al., 2019): 100 English literary texts with entity and event annotations.
- **TimeBank** (Pustejovsky et al., 2003): 183 English news articles containing annotations for events and temporal relations between them.

Both datasets follow similar guidelines for event annotation, with an important distinction: LitBank does not annotate events which have not occurred (eg: future, hypothetical or negated events). To overcome this gap, we remove all such events from TimeBank using available metadata about event modality and tense. Table 1 provides a brief overview of statistics for both datasets.

³Unlike prior work, we cannot use the ACE-2005 dataset since it tags specific categories of events, whereas we focus on tagging *all* possible events.

Model	In-Domain			Out-of-Domain		
	P	R	F1	P	R	F1
LSTM	70.7	78.4	74.4	23.5	75.2	35.8
LSTM-A	69.3	87.5	77.3	25.6	72.9	37.9
BiLSTM	75.4	76.3	75.9	27.6	68.8	39.4
BiLSTM-A	74.2	79.4	76.7	26.3	72.0	38.6
POS	77.4	81.1	79.2	26.4	79.8	39.6
POS-A	76.4	83.0	79.6	27.3	81.9	40.9
BERT	79.6	84.3	81.9	28.1	84.8	42.2
BERT-A	79.8	85.6	82.6	30.3	80.8	44.1

Table 3: Model performance on domain transfer experiments from TimeBank to LitBank. Presence of the -A suffix indicates that the model uses adversarial training.

3.2 Results and Analysis

Tables 2 and 3 present the results of our experiments. Table 2 shows the results when transferring from LitBank to TimeBank while Table 3 presents transfer results in the other direction. From Table 2 (transfer from LitBank to TimeBank), we see that ADA improves out-of-domain performance for all models, by 6.08 F1 on average. BERT-A performs best, reaching an F1 score of 49.6, using no labeled news data. Transfer experiments from TimeBank to LitBank (Table 3) showcase similar trends, with only BiLSTM not showing improvement with ADA. For other models, ADA results in an average out-of-domain F1 score improvement of 1.77. BERT-A performs best, reaching an F1 score of 44.1. We also note that models transferred from LitBank to TimeBank have high precision, while models transferred in the other direction have high recall. We believe this difference stems from the disparity in event density across corpora (Table 1). Since event density in LitBank is much lower, models transferred from LitBank tend to be slightly conservative (high precision), while models transferred from TimeBank are less so (high recall).

When transferring from LitBank to TimeBank, LSTM generalizes better than BiLSTM, which may be because BiLSTM has twice as many parameters making it more prone to overfitting. ADA gives a higher F1 boost with BiLSTM, indicating that it may be acting as a regularizer. Another interesting result is the poor performance of POS when transferring from LitBank to TimeBank. This might stem from the Stanford CoreNLP tagger (trained on news data) producing inaccurate tags for LitBank. Hence using automatically generated POS tags while training on LitBank does not produce

Category	%	Example
TimeBank Improvements		
Finance	54	the accord was unanimously approved
Political	12	the ukrainian parliament has already ratified it
Reporting	10	from member station kqed , auncil martinez reports
Law	10	mr. antar was charged last month in a civil suit
LitBank Improvements		
Archaic	6	his countenance became intolerably fervid
Animal Actions	6	the dogs left off barking , and ran about every way
Human Actions	18	a nod was the answer
Literary	14	there strikes the ebony clock

Table 4: Categorization of TimeBank and LitBank examples on which ADA shows improvement. Words in bold indicate events missed by BERT, but captured by BERT-A.

reliable POS embeddings.

On average, ADA makes supervised models more robust on out-of-domain data, with an average F1 score improvement of 3.9, at no loss of in-domain performance.

What cases does ADA improve on? To gain more insight into the improvements observed on using ADA, we perform a manual analysis of out-of-domain examples that BERT labels incorrectly, but BERT-A gets right. We carry out this analysis on 50 examples from TimeBank and LitBank each. We observe that an overwhelming number of cases from TimeBank use vocabulary in contexts unique to news (43/50 or 86%). This includes examples of financial events, political events and reporting events that are rarer in literature, indicating that ADA manages to reduce event extraction models’ reliance on lexical features. We make similar observations for LitBank though the proportion of improvement cases with literature-specific vocabulary is more modest (22/50 or 44%). These cases include examples with archaic vocabulary, words that have a different meaning in literary contexts and human/ animal actions, which are not common in news. Table 4 presents a detailed breakdown of

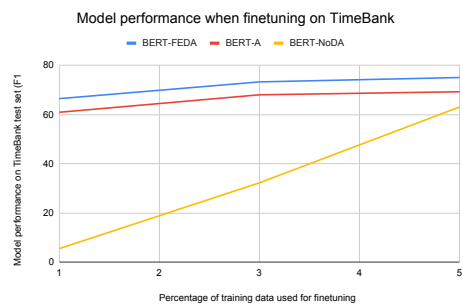


Figure 2: Improvement in model performance when finetuning on labeled training data from TimeBank

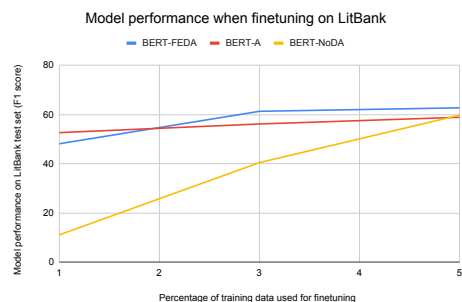


Figure 3: Improvement in model performance when finetuning on labeled training data from LitBank

these cases, along with examples.⁴

4 Incorporating Minimal Labeled Data

Finetuning on labeled data: We run finetuning experiments to study improvement in model performance on incorporating small amounts of labeled target domain data. For both domains, we finetune BERT-A, slowly increasing the percentage of labeled data used from 1%-5%.⁵ We compare BERT-A with two other models. The first model is naive BERT with no domain adaptation (BERT-NoDA). The second model is a BERT model trained via supervised domain adaptation (BERT-FEDA), which we use as an indicator of ceiling performance. The supervised domain adaptation method we use is the neural modification of frustratingly easy domain adaptation developed in Kim et al. (2016). Frustratingly easy domain adaptation (Daumé III, 2007) uses a feature augmentation strategy to improve performance when annotated data from both source and target domains is available. This algorithm simply duplicates input features 3 times,

⁴This table does not include generic improvement cases (i.e. no domain-specific vocabulary used), which formed 14% and 56% of improvement cases in TimeBank and LitBank.

⁵We run these experiments 5 times with different random subsets and average performance across all runs.

Dataset	P	R	F1
TimeBank	68.9	65.5	67.2
LitBank	40.3	71.5	51.5

Table 5: Model performance on both domains in the self-training paradigm

creating a source-specific, target-specific and general version of each feature. For source data, only the source-specific and general features are active, while only the target-specific and general features are active for target data. The neural modification works by duplicating the feature extractor module, which is the BiLSTM in our case.

Figures 2 and 3 present the results of these experiments. Performance of all models steadily improves with more data, but BERT-A starts with a much higher F1 score than BERT-NoDA, demonstrating that ADA boosts performance when little annotated training data is available. Performance increase of BERT-NoDA is surprisingly rapid, especially on LitBank. However, it is worth noting that 5% of the LitBank training set is $\sim 10,000$ tokens, which is a substantial amount to annotate. Therefore, BERT-A beats BERT-NoDA on sample efficiency. We can also see that BERT-A does not do much worse than BERT-FEDA, which performs supervised adaptation.

Using BERT-A to provide weak supervision:

We run further experiments to determine whether finetuned BERT-A can be leveraged for self-training (Yarowsky, 1995; Riloff and Wiebe, 2003). Self-training creates a teacher model from labeled data, which is then used to label a large amount of unlabeled data. Both labeled and unlabeled datasets are jointly used to train a student model. Algorithm 1 gives a quick overview of our self-training procedure. We use 1% of the training data as \mathcal{D}^l , with the remaining 99% used as \mathcal{D}^u . BERT-A acts as \mathcal{T} , while \mathcal{S} is a vanilla BERT model. Table 5 shows the results of self-training on both domains. Self-training improves model performance by nearly 7 F1 points on average. Increase on TimeBank is much higher which may be due to the high precision-low recall tendency of the teacher model.

5 Conclusion

In this work, we tackled the task of building generalizable supervised event trigger identification models using adversarial domain adaptation (ADA)

Algorithm 1 SelfTrain($\mathcal{D}^l, \mathcal{D}^u, \mathcal{T}$)

Input: Teacher Model \mathcal{T} , Labeled Data $\mathcal{D}^l = \{(x_1^l, e_1^l), \dots, (x_m^l, x_m^l)\}$, Unlabeled Data $\mathcal{D}^u = \{x_1^u, \dots, x_n^u\}$,

Output: Trained Student Model \mathcal{S}

- 1: Finetune the teacher model \mathcal{T} by minimizing cross-entropy loss on labeled data

$$\frac{1}{m} \sum_{i=1}^m \mathcal{L}(\mathcal{T}(x_i^l), e_i^l)$$

- 2: Generate labels $\{e_1^u, \dots, e_n^u\}$ for unlabeled data \mathcal{D}^u using \mathcal{T}
- 3: Train a student model \mathcal{S} by minimizing cross-entropy loss on both datasets $\mathcal{D}^l, \mathcal{D}^u$

$$\frac{1}{m} \sum_{i=1}^m \mathcal{L}(\mathcal{S}(x_i^l), e_i^l) + \frac{1}{n} \sum_{i=1}^n \mathcal{L}(\mathcal{S}(x_i^u), e_i^u)$$

- 4: Iterative training: Repeat step 2 using updated student model \mathcal{S}
-

to introduce domain-invariance. Our experiments with two domains (English literature and news) showed that ADA made supervised models more robust on out-of-domain data, with an average F1 score improvement of 3.9. Our best performing model (BERT-A) was able to reach 44-49 F1 across both domains using **no** labeled target domain data. Preliminary experiments showed that finetuning BERT-A on 1% labeled data, followed by self-training led to substantial improvement, reaching 51.5 and 67.2 F1 on literature and news respectively. While these results are encouraging, we are yet to match supervised in-domain model performance. Future directions to explore include incorporating noise-robust training procedures (Goldberger and Ben-Reuven, 2017) and example weighting (Dehghani et al., 2018) during self-training, and exploring lexical alignment methods from literature on learning cross-lingual embeddings.

Acknowledgements

This work was supported by the University of Pittsburgh Medical Center (UPMC) and Abridge AI Inc through the Center for Machine Learning and Health at Carnegie Mellon University. The authors would like to thank the anonymous reviewers for their helpful feedback on this work.

References

- Jun Araki and Teruko Mitamura. 2018. [Open-domain event detection using distant supervision](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 878–891, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. [Dense event ordering with a multi-pass architecture](#). *Transactions of the Association for Computational Linguistics*, 2:273–284.
- Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2018. [Adversarial deep averaging networks for cross-lingual sentiment classification](#). *Transactions of the Association for Computational Linguistics*, 6:557–570.
- Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu, and Jun Zhao. 2017. [Automatically labeled data generation for large scale event extraction](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 409–419, Vancouver, Canada. Association for Computational Linguistics.
- Naomi Daniel, Dragomir Radev, and Timothy Allison. 2003. [Sub-event based multi-document summarization](#). In *Proceedings of the HLT-NAACL 03 Text Summarization Workshop*, pages 9–16.
- Hal Daumé III. 2007. [Frustratingly easy domain adaptation](#). In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic. Association for Computational Linguistics.
- Mostafa Dehghani, Arash Mehrjou, Stephan Gouws, Jaap Kamps, and Bernhard Schölkopf. 2018. [Fidelity-weighted learning](#). In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.
- 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.
- George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. [The automatic content extraction \(ACE\) program – tasks, data, and evaluation](#). In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC’04)*, Lisbon, Portugal. European Language Resources Association (ELRA).
- James Ferguson, Colin Lockard, Daniel Weld, and Hananeh Hajishirzi. 2018. [Semi-supervised event extraction with paraphrase clusters](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 359–364, New Orleans, Louisiana. Association for Computational Linguistics.
- Yaroslav Ganin and Victor S. Lempitsky. 2015. [Unsupervised domain adaptation by backpropagation](#). In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1180–1189. JMLR.org.
- Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. 2016. [Domain-adversarial training of neural networks](#). *J. Mach. Learn. Res.*, 17:59:1–59:35.
- Jacob Goldberger and Ehud Ben-Reuven. 2017. [Training deep neural-networks using a noise adaptation layer](#). In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- Ralph Grishman and Beth Sundheim. 1996. [Message understanding conference- 6: A brief history](#). In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.
- Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare R. Voss, Jiawei Han, and Avirup Sil. 2016. [Liberal event extraction and event schema induction](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 258–268, Berlin, Germany. Association for Computational Linguistics.
- Ruihong Huang and Ellen Riloff. 2012. [Bootstrapped training of event extraction classifiers](#). In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 286–295, Avignon, France. Association for Computational Linguistics.
- Katherine Keith, Abram Handler, Michael Pinkham, Cara Magliozzi, Joshua McDuffie, and Brendan O’Connor. 2017. [Identifying civilians killed by police with distantly supervised entity-event extraction](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1547–1557, Copenhagen, Denmark. Association for Computational Linguistics.
- Jin-Dong Kim, Tomoko Ohta, and Jun’ichi Tsujii. 2008. [Corpus annotation for mining biomedical events from literature](#). *BMC Bioinform.*, 9.
- Young-Bum Kim, Karl Stratos, and Ruhi Sarikaya. 2016. [Frustratingly easy neural domain adaptation](#).

- In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 387–396, Osaka, Japan. The COLING 2016 Organizing Committee.
- Zheng Li, Yu Zhang, Ying Wei, Yuxiang Wu, and Qiang Yang. 2017. [End-to-end adversarial memory network for cross-domain sentiment classification](#). In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 2237–2243. ijcai.org.
- Shasha Liao and Ralph Grishman. 2010. [Filtered ranking for bootstrapping in event extraction](#). In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 680–688, Beijing, China. Coling 2010 Organizing Committee.
- Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. [Adversarial multi-task learning for text classification](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–10, Vancouver, Canada. Association for Computational Linguistics.
- James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. 2003. The timebank corpus. In *Corpus linguistics*, volume 2003, page 40. Lancaster, UK.
- Ellen Riloff and Janyce Wiebe. 2003. [Learning extraction patterns for subjective expressions](#). In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 105–112.
- Roser Saurí, Robert Knippen, Marc Verhagen, and James Pustejovsky. 2005. [Evita: A robust event recognizer for QA systems](#). In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 700–707, Vancouver, British Columbia, Canada. Association for Computational Linguistics.
- Darsh Shah, Tao Lei, Alessandro Moschitti, Salvatore Romeo, and Preslav Nakov. 2018. [Adversarial domain adaptation for duplicate question detection](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1056–1063, Brussels, Belgium. Association for Computational Linguistics.
- Matthew Sims, Jong Ho Park, and David Bamman. 2019. [Literary event detection](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3623–3634, Florence, Italy. Association for Computational Linguistics.
- David Yarowsky. 1995. [Unsupervised word sense disambiguation rivaling supervised methods](#). In *33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, Cambridge, Massachusetts, USA. Association for Computational Linguistics.
- Jianfei Yu, Minghui Qiu, Jing Jiang, Jun Huang, Shuangyong Song, Wei Chu, and Haiqing Chen. 2018. [Modelling domain relationships for transfer learning on retrieval-based question answering systems in e-commerce](#). In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, pages 682–690. ACM.
- Quan Yuan, Xiang Ren, Wenqi He, Chao Zhang, Xinhe Geng, Lifu Huang, Heng Ji, Chin-Yew Lin, and Jiawei Han. 2018. [Open-schema event profiling for massive news corpora](#). In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 587–596. ACM.
- Ying Zeng, Yansong Feng, Rong Ma, Zheng Wang, Rui Yan, Chongde Shi, and Dongyan Zhao. 2018. [Scale up event extraction learning via automatic training data generation](#). In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 6045–6052. AAAI Press.

Appendix

A Implementation Details

All models are implemented in PyTorch. We use 300-dimensional GloVe embeddings while training on TimeBank and 100-dimensional Word2Vec embeddings trained on Project Gutenberg texts (similar to (Sims et al., 2019)) while training on LitBank. Both source and target domains share a common vocabulary and embedding layer which is not finetuned during the training process. All LSTM models use a hidden size of 100, with an input dropout of 0.5. The POS model uses 50-dimensional embeddings for POS tags which are randomly initialized and finetuned during training. The BERT model uses the uncased variant of BERT-Base for feature extraction. We generate token representations by running BERT-Base and concatenating the outputs of the model’s last 4 hidden layers. The event classifier is a single-layer 100-dimensional MLP. For the adversarial training setup, we experiment with values from [0.1,0.2,0.5,1.0,2.0,5.0] for the hyperparameter λ . The domain predictor (adversary) is a 3-layer MLP with each layer having a dimensionality of 100 and ReLU activations between layers. We train all models with a batch size of 16 and use the Adam optimizer with default learning rate settings. Models are trained for 1000 epochs, with early stopping. For finetuning experiments, we train for 10 epochs.