Open-Domain Event Detection using Distant Supervision

Jun Araki and **Teruko Mitamura**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
junaraki@cs.cmu.edu teruko@andrew.cmu.edu

Abstract

This paper introduces *open-domain event detection*, a new event detection paradigm to address issues of prior work on restricted domains and event annotation. The goal is to detect all kinds of events regardless of domains. Given the absence of training data, we propose a distant supervision method that is able to generate high-quality training data. Using a manually annotated event corpus as gold standard, our experiments show that despite no direct supervision, the model outperforms supervised models. This result indicates that the distant supervision enables robust event detection in various domains, while obviating the need for human annotation of events.

1 Introduction

Events are a key semantic component integral to natural language understanding. They are a ubiquitous linguistic phenomenon appearing in various domains. For clarification, we use the term 'domain' to refer to a specific genre of text, such as biology, finance, and so forth. Prior studies on automatic event detection traditionally focus on limited types of events, mainly defined by several research initiatives and shared tasks in a few domains:

- Newswire: TIPSTER (Onyshkevych et al., 1993) and MUC (Grishman and Sundheim, 1996)
- Multi-domain: ACE (Doddington et al., 2004) and TAC KBP (Mitamura et al., 2016).
- Biology: PASBio (Wattarujeekrit et al., 2004), GENIA (Kim et al., 2008), BioNLP (Kim et al., 2009) and ProcessBank (Berant et al., 2014).

Although closed-domain event detection might be of practical use in some domain-specific scenarios, it only contributes to partial understanding of events because it only addresses a subset of events by definition. On the other hand, there is an established consensus that in order to advance natural language applications such as open-domain question answering, we need automatic event identification techniques with larger, wider, and more consistent coverage (Saurí et al., 2005; Pradhan et al., 2007).

There are several pieces of prior work on open-domain events, but they have some limitations with respect to coverage of events. Lexical databases such as WordNet (Miller et al., 1990), FrameNet (Baker et al., 1998) and PropBank (Palmer et al., 2005) can be viewed as a superset of events, and their subtaxonomies seem to provide an extensional definition of events. However, these databases have a narrow coverage of events because they are generally expected to cover basic terminology due to their dictionary nature. For instance, none of WordNet, FrameNet and PropBank covers current terminology and proper nouns such as 'Hurricane Katrina'. OntoNotes (Weischedel et al., 2011) is aimed at covering an unrestricted set of events and entities, but its event annotation is limited to a small number of eventive nouns. TimeML (Pustejovsky et al., 2003) annotates events and times in a domain-agnostic manner, focusing on temporal aspects of events, but it does not deal with multi-word and generic events. ECB+ (Cybulska and Vossen, 2014) augments the extended EventCorefBank corpus (Lee et al., 2012) by re-annotating events and event coreference. Our initial analysis shows that it annotates events only in a portion of each document, not all events in the text. Ritter et al. (2012) address open-domain event detection on Twitter. Their annotation follows TimeML and thus is likely to have similar problems to TimeML. Richer

Proceedings of the 27th International Conference on Computational Linguistics, pages 878–891
Santa Fe, New Mexico, USA, August 20-26, 2018.

Event Description (RED) (Palmer et al., 2016) defines events in a general manner, but its annotation was performed only in the clinical domain (O'Gorman et al., 2016).

To address these issues, we introduce a new paradigm of *open-domain event detection*. Our goal is to detect all kinds of events without a specific event type, while not limiting events to particular domains and syntactic types. As compared to the traditional closed-domain event detection described above, our target is more challenging because there is no unified definition of events with wider and more consistent coverage, and there is no training data. Due to the ubiquity and ambiguities of events, human annotation of events in the open domain is substantially expensive, ending up with a small dataset which does not make supervised models generalize well. In this work, we propose a novel approach to circumvent the data sparsity problem in open-domain event detection. At the core of the approach is distant supervision from WordNet (Miller et al., 1990). Our main hypothesis is that the distant supervision allows us to build models detecting events robustly in various domains, while obviating the need for human annotation of events. We verify this hypothesis in both heuristics-based and classifier-based algorithms.

Our contribution is as follows:

1. We introduce **a new paradigm of open-domain event detection**, whose goal is to detect all kinds of events. Specifically, the differences from prior work are events in unrestricted domains and with wider coverage.
2. Our **distant supervision method** is able to generate high-quality training data (see Section 5.1). The method is not bounded to any particular datasets, offering a versatile solution for event detection. Using a manually annotated corpus as gold standard, our experiments show that despite no direct supervision, the distantly supervised model outperforms supervised models in both in-domain and out-domain settings (see Section 5.3).
3. To facilitate future studies on event detection, we **release the new corpus**[1] of human-annotated events in 10 different domains such as geology and economics. The annotated events comprise verbs, nouns, adjectives, and phrases which are continuous or discontinuous (see Section 2.2). Despite this relatively wide and flexible annotation of events, we achieved high inter-annotator agreement (see Section 4).

2 Definition of Events

We give our definition of events from both semantic and syntactic perspective.

2.1 Semantic Perspective: Eventualities

Events are a highly ambiguous notion, and thus there are many ways to define them. In this work, we use the notion of eventualities by Bach (1986) and define the three classes on the basis of durativity and telicity (Moens and Steedman, 1988; Pulman, 1997):

- **states**: notions that are durative and changeless, e.g, want, own, love, resemble
- **processes**: notions that are durative and atelic, e.g., walking, sleeping, raining
- **actions**[2]: notions that are telic or momentaneous happenings, e.g., build, walk to Boston, recognize

We define expressions as events if they denote an eventuality, i.e., a state, a process, or an action.

2.2 Syntactic Perspective: Event Nuggets

From a syntactic perspective, we use the notion of event nugget (Mitamura et al., 2015), which is a semantically meaningful unit that expresses an event. We give several examples below, where we use boldface to highlight event nuggets and underlines to show units of multi-word ones.

(1) He **opened fire** at the teller in the bank.
(2) I **cried** when my grandpa **kicked** the **bucket**.
(3) Tom was **happy** when he **received** a present.
(4) Susan **turned** the TV **on**.
(5) She **responded** his email **dismissively**.

[1] https://bitbucket.org/junaraki/coling2018-event

[2] Bach (1986) uses the term 'events' to refer to this class. In this work, we use 'actions' instead for clarification.

Event nuggets can be either a single word (verb, noun, or adjective) or a phrase which is continuous or discontinuous. For example, 'turned ... on' in (2) is a discontinuous phrasal event nugget, excluding 'the TV'. For more details, we refer readers to our annotation guidelines.[3]

3 Distantly Supervised Event Detection

This section describes our approach for open-domain event detection. Figure 1 shows a high-level overview of the approach. As shown, the algorithm comprises two phases: training data generation and event detection. At the core of the approach is distant supervision from WordNet[4] in the former phase to address ambiguities on eventiveness and generate high-quality training data automatically.

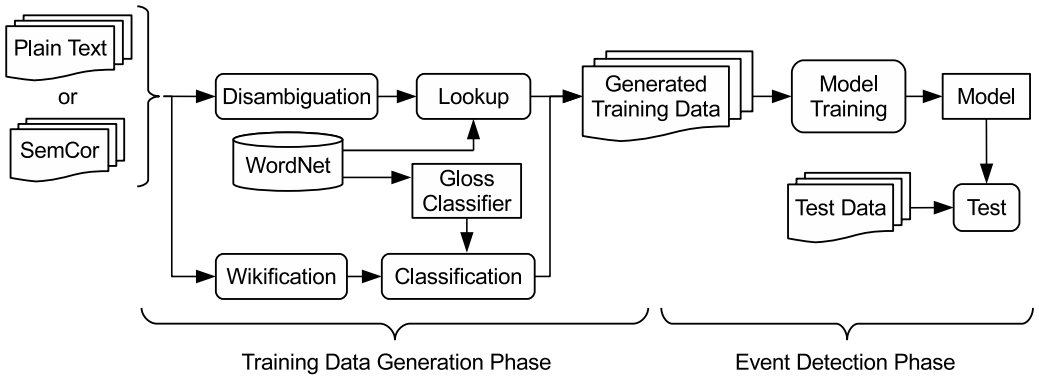

Figure 1: An overview of our distantly-supervised open-domain event detection.

3.1 Training Data Generation

The goal of training data generation is to generate high-quality training data automatically from unannotated text. We first disambiguate text using a WordNet-based word sense disambiguation tool. Given sense-annotated text as input, we implement a rule-based algorithm, which we refer to as **RULE**. It is our basis for generating training data. Looking at our annotation guidelines (see Section 4), we employ several heuristics to identify event nuggets, according to syntactic types:

Verbs. We detect most of main verbs as eventive, excluding be-verbs and auxiliary verbs. However, some exceptions exist. In the examples below, we use italic face to highlight non-eventives.

(6) That is what I **meant**.
(7) 'Enormous' *means* 'very big'.

In (6), 'meant' is eventive because it indicates the action of "intend to say" whereas 'means' in (7) is not because it merely shows equality, playing the almost same role as a be-verb. Thus, we define a set of non-eventive verb senses and filter out verbs if their disambiguated sense is in the set.

Nouns. There are also ambiguous nouns:

(8) His **payment** was late.
(9) His *payment* was $10.
(10) **Force** equals mass times acceleration.

In (8), 'payment' is eventive because it means the action of his paying something while 'payment' in (9) is not because it refers to specific money paid by him, which is $10. These examples also show that eventive nouns cannot be simply approximated by verb nominalizations. 'Force' in (10) can be easily disambiguated to its physical sense, but still deciding its eventiveness is difficult. To address the issue, we make use of distant supervision from WordNet. Let word_n^i denote the i-th sense (synset) of noun word in WordNet. For example, car_n^1 is the first synset of noun 'car'. Looking at textual definitions of synsets called *glosses*, we assume that there is a semantic correspondence between the components of eventualities described in Section 2.1 and the following WordNet synsets:

- state_n^2: the way something is with respect to its main attributes

[3]We release the guidelines at `https://bitbucket.org/junaraki/coling2018-event`.
[4]We access WordNet 3.0 using NLTK (Bird et al., 2009).

- process^{6}_{n}: a sustained phenomenon or one marked by gradual changes through a series of states
- event^{1}_{n}: something that happens at a given place and time

We detect nouns as events if their disambiguated sense is subsumed by the three synsets above through (instance-)hyponym relations.

Adjectives. Adjectives can also be ambiguous:

(11) Mary was **talkative** at the **party**.

(12) Mary is a *talkative* person.

In (11), 'talkative' is eventive because it implies that Mary talked a lot at the party, whereas 'talkative' in (12) is not because it just indicates Mary's personal attribute. As illustrated, major problems with adjectives are to differentiate states from attributes and to figure out if they imply actual occurrences (Palmer et al., 2016). Unlike nouns, no direct supervision is available from WordNet because it does not have hyponym taxonomies for adjectives. Thus, we use simple and conservative heuristics to detect adjectives as events if they are originated from present and past participles of verbs, illustrated as follows:

(13) There is a **man-made** river in the country.

(14) The tower has 20,000 **sparkling** lights.

We convert these adjectives to verbs in the infinitive form using pattern.en (De Smedt and Daelemans, 2012).

Adverbs. We develop heuristics using WordNet. We first convert adverbs to adjectives by looking up pertainyms (relational adjectives) and seeking verbs derivationally related to the adjectives in WordNet. Adverbs connect with their modifying verbs, forming a single event nugget, as illustrated in (5). Thus, we combine eventive adverbs with such verbs to detect resulting verb phrases as events.

Phrases. Following Schneider et al. (2014), we define *phrases* to be lexicalized combinations of two or more words that are exceptional enough to be considered as single units in the lexicon. We assume that this definition is suitable to event detection because the exceptionality of multi-word units in the phrase lexicon translates to the meaningfulness of textual units of (phrasal) event nuggets. From the perspective of open-domain event detection, supervised phrase detection models are likely suboptimal because they might be limited to particular domains or overfitting to small datasets. Therefore, we explore a simple dictionary-lookup approach to detect WordNet phrasal verbs as events, inspired by Yin and Schütze (2015). One enhancement to their approach is that we examine dependencies using Stanford CoreNLP (Manning et al., 2014), illustrated as follows:

(15) Snipers were **picking** them **off**.

(16) He **picked** an apple off the tree.

In (15), 'picking ... off' forms a discontinuous phrasal verb, whereas 'picked' in (16) does not. Dependencies can be of help to resolve these two cases. In the former case, a dependency relation 'picking $\xrightarrow{\texttt{compound:prt}}$ off' is a direct signal of the phrasal verb construction. As for noun phrases, we apply our heuristics for nouns to head words of the phrases.

3.2 Enhancements with Wikipedia

This subsection describes two techniques to enhance our training data generation: heuristics-based enhancement (Section 3.2.1) and classifier-based enhancement (Section 3.2.2). One disadvantage of RULE is the limited coverage of WordNet. As described in Section 1, WordNet does not cover many proper nouns that we generally see in newspaper articles, such as the following:

(17) Property damage by **Hurricane Katrina** around $108 billion.

(18) The **Cultural Revolution** was ...

In order to achieve higher recall, we incorporate Wikipedia to capture proper nouns which are not in WordNet, motivated by the fact that Wikipedia has a much broader coverage of concepts than WordNet synsets.[5] We use the Illinois Wikifier (Ratinov et al., 2011) to extract Wikipedia concepts from text.

[5]WordNet 3.0 has 120K synsets, and English Wikipedia has 5.5M articles as of October 2017, as shown at `https://stats.wikimedia.org/EN/TablesWikipediaEN.htm`.

3.2.1 Heuristics-based Enhancement

For our first enhancement, we make two assumptions: (1) the first sentence of a Wikipedia article provides a high-quality gloss of its corresponding concept, and (2) the syntactic head of a gloss represents a high-level concept carrying significant information to decide eventiveness. The first assumption is supported by Wikipedia's style manual on how to write the first sentence of an article.[6] The manual says "If an article's title is a formal or widely accepted name for the subject, display it ... as early as possible in the first sentence." For instance, the first sentence of entry Electron is:

(19) The electron is a subatomic particle with a negative elementary electric charge.

The gloss of Electron is the underlined text above. Our analysis shows that most Wikipedia articles follow the first-sentence format.

The second assumption is illustrated by the syntactic head of the Electron gloss, which is 'particle'. Based on the assumptions, we develop head-based heuristics, which we call **HeadLookup**. We find the syntactic head of a gloss using dependencies and disambiguate the head using a state-of-the-art word sense disambiguation tool IMS (It Makes Sense) (Zhong and Ng, 2010). We then check if the head's sense is subsumed by the three synsets of state^2_n, $\mathsf{process}^6_n$, and event^1_n. In the case of Electron, the head's sense atom^2_n is not under the synsets. Thus, the model concludes that Electron is non-eventive. Note that HeadLookup itself is a general technique which can be applied to any gloss. Our first enhancement applies it to Wikipedia glosses, and we refer to the enhanced model as **RULE-WP-HL**.

3.2.2 Classifier-based Enhancement

Our second enhancement leverages a binary gloss classifier to decide the eventiveness of proper nouns. We refer to this enhanced model as **RULE-WP-GC**. We use WordNet glosses to train the classifier. Our assumption is that although WordNet and Wikipedia are maintained by different people for different purposes, the classifier trained on WordNet glosses generalizes well against unseen Wikipedia concepts because glosses of the two resources have comparable quality.

Data Collection from WordNet. We collect our gloss datasets automatically from WordNet. The goal of data collection is to create a large amount of dataset $D = D_+ \cup D_-$ where D_+ is a set of eventive (positive) glosses, D_- is a set of non-eventive (negative) ones, and $D_+ \cap D_- = \emptyset$. Since WordNet provides a gloss for each synset, the goal reduces to creating a set of positive synsets S_+ and a set of negative ones S_-. Given root synset s, we collect a subset of synsets S_s (including s) by traversing the WordNet taxonomy under s through hyponym and instance-hyponym relations. Using the three synsets introduced in Section 3.1, we have $S_+ = S_{\mathsf{event}^1_n} \cup S_{\mathsf{state}^2_n} \cup S_{\mathsf{process}^6_n}$. With respect to S_-, we simply take all the WordNet synsets that are not in S_+. Table 1 gives several examples of glosses in D_+ and D_-. As shown, the ambiguous word 'payment' has positive and negative synsets. The sizes of D_+ and D_- are $|D_+| = 13,415$ and $|D_-| = 68,700$.

Synset	Gloss
riding^2_n	travel by being carried on horseback
shower^3_n	a brief period of precipitation
$\mathsf{payment}^2_n$	the act of paying money
pork^1_n	meat from a domestic hog or pig
year^1_n	a period of time containing 365 (or 366) days
$\mathsf{payment}^1_n$	a sum of money paid or a claim discharged

Table 1: Examples of WordNet glosses in D. The examples above and below the dashed line are from D_+ and D_-, respectively.

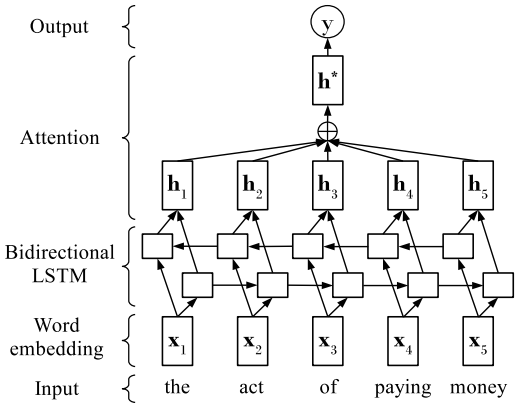

Figure 2: The bidirectional LSTM model with a self-attention mechanism.

[6] https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Lead_section\#First_sentence

Learning. We train the binary classifier on D using a bidirectional long short-term memory (BLSTM) (Graves and Schmidhuber, 2005). We call the classifier **GC-BLSTM**. Given an input vector $\mathbf{x}_t$ at time step t, let us denote the forward hidden state as $\overrightarrow{\mathbf{h}_t}$ and the backward one as $\overleftarrow{\mathbf{h}_t}$. We obtain the hidden state of a BLSTM using concatenation: $\mathbf{h}_t = [\overrightarrow{\mathbf{h}_t}; \overleftarrow{\mathbf{h}_t}]$. We then use a linear projection of $\mathbf{h}_T$ into two classes: $y = 1$ (eventive) and $y = 0$ (non-eventive). Finally, we add a softmax layer on top of the linear projection, and train the model using the binary cross-entropy loss.

Attention. Neural networks with attention mechanisms have achieved great success in a wide variety of natural language tasks. The basic idea is to enable the model to attend to all past hidden vectors and put higher weights on important parts so that the model can encode the sequence information more effectively. This idea intuitively makes sense for gloss classification as well, because syntactic heads are likely more important, as illustrated in Section 3.2.1. As shown in Figure 2, we leverage a self-attention mechanism, following (Zhou et al., 2016; Lin et al., 2017). Let $\mathbf{H} \in \mathbb{R}^{d \times T}$ denote a matrix comprising hidden vectors $[\mathbf{h}_1, \ldots, \mathbf{h}_T]$ where d is the dimensionality of a hidden vector. The self-attention mechanism computes the hidden state as follows:

$$\mathbf{M} = \tanh(\mathbf{H}) \tag{1}$$

$$\alpha = \text{softmax}(\mathbf{w}^T \mathbf{M}) \tag{2}$$

$$\mathbf{r} = \mathbf{H}\alpha^T \tag{3}$$

$$\mathbf{h}^* = \tanh(\mathbf{r}) \tag{4}$$

where a vector $\alpha \in \mathbb{R}^T$ is attention weights and $\mathbf{w} \in \mathbb{R}^d$ is a parameter vector. We refer to the attention-based classifier as **GC-BLSTM-Attn**.

Implementation Details. We randomly sample 1,000 examples from each of D_+ and D_- to create a test set and a validation set, and use the rest of D for a training set. We use the 300-dimensional GloVe vectors[7] from (Pennington et al., 2014) and do not fine-tune them during training. We map all out-of-vocabulary words to a single vector randomly initialized by uniform sampling from $[-0.01, 0.01]$. We use a single hidden layer of 100 dimensions, i.e., $d = 100$. We optimize model parameters using minibatch stochastic gradient descent (SGD) with momentum 0.9. We choose an initial learning rate of $\eta_0 = 1.0\text{e}{-3}$. We use a minibatch of size 1. To mitigate overfitting, we apply the dropout method (Srivastava et al., 2014) to the inputs and outputs of the network. We also employ L2 regularization. We perform a small grid search over combinations of dropout rates $\{0.0, 0.1, 0.2\}$ and L2 regularization penalties $\{0.0, 1.0\text{e}{-3}, 1.0\text{e}{-4}\}$. We use early stopping based on performance on the validation set.

3.3 Learning for Event Detection

After generating training data, we train a supervised event detection model on the synthesized data. As seen in the self-training model by Liao and Grishman (2011), erroneously generated training data worsen system performance. Therefore, we need to generate training data as accurately as possible. On the other hand, our algorithm for generating training data comprises at least three non-trivial (error-prone) submodules: disambiguation, wikification, and gloss classification. To eliminate negative effects of disambiguation errors, we choose the SemCor corpus (Miller et al., 1993) as our base text for training data generation. SemCor has human-annotated WordNet senses on 186 documents in numerous genres. We apply our rule-based event detector to generate training data automatically from SemCor.

We formalize event detection as a sequence labeling problem and employ a BLSTM for sequence modeling. One difference from traditional sequence labeling problems is that our output includes discontinuous phrases. Thus, we leverage an extended version of the standard BIO tagging scheme, inspired by Metke-Jimenez and Karimi (2016). Besides the three tags of B, I and O, we introduce two additional tags: DB and DI. DB means the beginning of a discontinuous concept, and DI the continuation of a discontinuous concept. The BLSTM model computes a hidden representation from each input word and then predicts one of $\{\text{B}, \text{I}, \text{DB}, \text{DI}, \text{O}\}$. For the BLSTM model, we use a fixed concatenation of the GloVe embeddings and 50-dimensional word embeddings from Turian et al. (2010) under the same assumption

[7] https://nlp.stanford.edu/projects/glove/

as (Lee et al., 2017) that different learning objectives of the GloVe and Turian embeddings can provide orthogonal information. We additionally employ 10-dimensional part-of-speech embeddings. We train the model with the objective of minimizing cross-entropy loss. We use early stopping based on the loss on a validation set.

4 Open-Domain Event Corpus

Since our target is unrestricted domains, we need gold standard data to evaluate whether systems work robustly in various domains. However, annotating events manually in all domains would be unrealistic due to annotation cost. To make the corpus creation manageable while retaining the domain diversity, we select 100 articles in Simple English Wikipedia, comprising 10 from each of 10 different domains. The domains are: architecture, chemistry, disasters, diseases, economics, education, geology, history, politics, and transportation. We choose Simple Wikipedia because our annotators are not necessarily experts in these domains, and the simplified sentences could facilitate our annotation of events against text from the wide variety of domains. Three annotators perform annotation independently following our annotation guidelines, and we measure inter-annotator agreement using the pairwise F1 score under two conditions: strict match and partial match. The former checks whether two annotations have exactly the same span. The latter checks whether there is an overlap between annotations, with the restriction that each annotation can only be matched to one annotation by the other annotator. We compute the agreements for all annotator pairs and average them for the overall agreement. As a result, the inter-annotator agreement was 80.7% (strict match) and 90.3% (partial match). Finally, the most experienced annotator finalizes event annotation. We refer to the event corpus as **SW100**.[8]

Table 2(a) shows that event nuggets appear in the 10 domains almost uniformly, ensuring the ubiquity of events. We use Stanford CoreNLP to tokenize text and decide head words of event nuggets with dependencies. Table 2(b) counts multi-word event nuggets by part-of-speech tags of their heads, amounting to 955. 24% of the 955 are discontinuous, and most (97%) of the discontinuous multi-word event nuggets are verb phrases. 'Others' in Table 2(b) include pronouns, demonstrative determiners, and numbers.

Domain	# (%)	Domain	# (%)
Architecture	475 (8.8)	Education	653 (12.1)
Chemistry	576 (10.7)	Geology	483 (8.9)
Disaster	510 (9.4)	History	486 (9.0)
Disease	618 (11.4)	Politics	534 (10.0)
Economics	479 (8.9)	Transportation	583 (10.8)

(a) Event nuggets with respect to domains.

	Single-word	Multi-word	All
Verb	2799 (51.9)	560 (10.4)	3359 (62.2)
Noun	1273 (23.6)	382 (7.1)	1655 (30.6)
Adjective	192 (3.6)	2 (0.0)	194 (3.6)
Others	178 (3.3)	11 (0.2)	189 (3.5)
All	4442 (82.3)	955 (17.7)	5397 (100.0)

(b) Event nuggets with respect to syntactic types.

Table 2: Corpus statistics of SW100. Percentages (%) are shown in parentheses.

5 Experimental Results

This section describes our experimental results.

5.1 Results of Gloss Classification

Gloss classification is a binary classification subtask for training data generation, aimed to improve event detection by capturing proper nouns with Wikipedia knowledge. We use two datasets to evaluate our gloss classifiers described in Section 3.2. One is the test set of WordNet glosses described in Section 3.2.2. However, we actually care about the gloss classification performance against unseen Wikipedia concepts. Thus, we create an additional dataset which comprises Wikipedia concepts that do not appear in WordNet. We collect 100 eventive and 100 non-eventive Wikipedia glosses in 10 domains[9], independently of SW100. We measure the performance of gloss classification using accuracy. For comparison, we consider two baselines. **BoW-LR** is a bag-of-words model trained with logistic

[8] https://bitbucket.org/junaraki/coling2018-event

[9] Economics, history, politics, psychology, architecture, earth science, physics, chemistry, biology, and medicine.

regression, and **DAN** is a deep average network proposed by Iyyer et al. (2015). Since DAN can be seen as a neural bag-of-words model, these two models are word-order insensitive baselines. Table 3 shows that GC-BLSTM-Attn performs best and significantly better than GC-BLSTM on both WordNet and Wikipedia datasets. This result verifies our assumptions that the classifier trained on WordNet glosses generalizes well against unseen Wikipedia concepts and that the attention mechanism is effective for gloss classification.

Model	WordNet	Wikipedia
HeadLookup	77.80	73.50
BoW-LR	79.50	73.00
DAN	83.15	64.00
GC-BLSTM	90.10	80.00
GC-BLSTM-Attn	**91.65****	**85.00***

Table 3: Accuracy of gloss classifiers. The stars indicate statistical significance compared to GC-BLSTM (*: $p < 0.05$; **: $p < 0.005$) based on McNemar's test.

Model	Strict			Partial		
	P	R	F1	P	R	F1
VERB (Baseline)	79.5	51.7	62.7	95.4	62.0	75.2
PRED (Baseline)	55.1	62.4	58.5	67.6	76.6	71.8
RULE	80.1	77.0	78.5	89.0	85.5	87.2
RULE-WP-HL	80.5	77.5	79.0	88.6	85.3	86.9
RULE-WP-GC	80.8	77.7	**79.2**	89.1	85.7	**87.3**

Table 4: Performance of the rule-based event detectors on SW100.

5.2 Results of Training Data Generation

We measure the performance of the rule-based event detectors on SW100 using precision (P), recall (R), and F1 with the two matching options described in Section 4. We use IMS (It Makes Sense) for disambiguation. Table 4 shows the results. **VERB** is a simple baseline that detects all single-word main verbs as events, excluding be-verbs and auxiliary verbs. **PRED** is another baseline that detects all predicates as events by running a state-of-the-art semantic role labeler called PathLSTM[10] (Roth and Lapata, 2016). Since PathLSTM is trained on both PropBank and NomBank, it is able to detect both verbal and nominal predicates. However, semantic role labeling has a different focus on predicate-argument structures. More specifically, the combination of PropBank and NomBank has a narrower coverage of events while having non-event predicates. This difference explains PathLSTM's relatively low performance of 58.5 strict F1, even underperforming the VERB baseline. The performance difference between RULE and VERB mostly comes from nouns, indicating that our WordNet-based heuristics is effective. We found that RULE-WP-GC achieves the best F1. This result shows that the proposed enhancements with Wikipedia can contribute to generating higher-quality training data.

We then applied the best-performing RULE-WP-GC to SemCor and found that the generated data contains 59,796 event nuggets in total. We randomly split this data or its subset to 9:1 with respect to the number of documents, creating training and validation data. We train the neural event detector described in Section 3.3 on the training data and measure its performance on SW100. Figure 3 shows how the amount of training data affects the performance of event detection. As shown, a larger amount of training data enables the model to achieve better performance.

5.3 Comparison with Supervised Models

In order to test the robustness of our distant supervision model, in this experiment we divide SW100 into in-domain and out-domain datasets by randomly sampling 5 domains for each. We further split the in-domain dataset into training (60%), validation (20%) and test subsets (20%) with respect to the number of documents, and then train the BLSTM event detection model described in Section 3.3. We repeat this procedure three times and measure the average of F1 scores with strict match (Table 5). We refer to the model trained on the in-domain data as **BLSTM** and the model trained on the data generated from SemCor as **DS-BLSTM**. In all the three runs, DS-BLSTM outperforms BLSTM in both in-domain and out-domain settings. The performance difference in the out-domain setting is statistical significant at $p < 0.05$ based on a two-tailed t-test. BLSTM ends up overfitting to the training data, and its weak generalization power is more evident in the out-domain setting. In contrast, DS-BLSTM performs robustly in both settings. Table 6 shows more detailed performance of DS-BLSTM.

[10] https://github.com/microth/PathLSTM

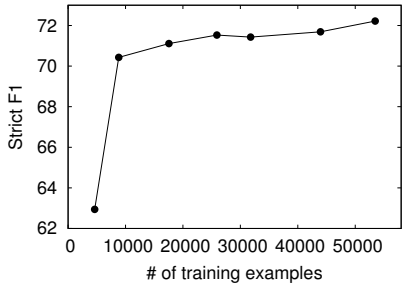

Figure 3: Performance of the event detection model on SW100 with respect to the number of training examples generated from SemCor.

Setting	Model	Strict F1	Partial F1
In-domain	BLSTM	73.8	85.9
	DS-BLSTM	**76.1**	**88.0**
Out-domain	BLSTM	67.9	82.8
	DS-BLSTM	**71.3**	**86.6**

Table 5: Results of event detection.

Domain	Strict			Partial		
	P	R	F1	P	R	F1
Architecture	81.1	71.2	75.8	92.3	81.1	86.3
Chemistry	75.5	71.2	73.3	91.0	85.8	88.3
Disaster	80.7	70.6	75.3	95.7	83.7	89.3
Disease	66.6	53.2	59.2	89.9	71.8	79.9
Economics	73.7	67.8	70.7	93.2	85.8	89.3
Education	71.8	67.4	69.5	86.9	81.6	84.2
Geology	78.6	71.6	75.0	92.3	84.1	88.0
History	77.4	69.8	73.4	92.2	83.1	87.4
Politics	78.2	69.7	73.7	93.5	83.3	88.1
Transportation	81.5	75.6	78.5	91.5	84.9	88.1

(a) With respect to domains.

Syntactic type	Strict			Partial		
	P	R	F1	P	R	F1
Verbs	79.7	83.3	81.5	94.9	99.2	97.0
Nouns	67.4	51.7	58.5	82.5	63.4	71.7
Adjectives	68.9	26.3	38.1	68.9	26.3	38.1

(b) With respect to syntactic types.

Table 6: Detailed performance of DS-BLSTM.

6 Error Analysis and Discussion

This section discusses our error analysis.

Analysis of Gloss Classification. One error pattern is that the gloss of an eventive example is partially or completely overlapped with that of non-eventive one, confusing the classifier:

- broadcast^2_n: a radio or television show
- laugh_track^1_n: prerecorded laughter added to the soundtrack of a radio or television show

The former is eventive and the latter is not. Beside such errors, we found possible inconsistencies in WordNet entries in terms of eventiveness:

- sufficiency^1_n: sufficient resources to provide comfort and meet obligations
- unanimity^1_n: everyone being of one mind
- minority^3_n: any age prior to the legal age

The first three synsets are in the state^2_n taxonomy, but the ways of defining them, especially the syntactic heads of glosses (underlined above) sound like non-eventive entities rather than events.

#	Submodule	Description	Examples
(1)	Phrase detection	A phrase is missing or incorrectly detected.	amount to, stay clear, take one's toll
(2)	Wikification	A proper name is not identified or disambiguated into an incorrect entry in Wikipedia.	Polish Revolution, Battle of The Little Horn
(3)	Wikipedia gloss extraction	A corresponding Wikipedia article does not provide a gloss of an expected form.	Spanish flu, Exxon Valdez oil spill
(4)	Gloss classification	A disambiguated gloss is misclassified.	Anglican parson, Archbishop

Table 7: Noticeable errors of our training data generation.

Analysis of Training Data Generation. Besides gloss classification, our training data generation is subject to errors from the rule-based event detection and wikification. Table 7 shows noticeable errors in training data generation. The cause of error (1) is small coverage of WordNet phrases, particularly verb phrases. Error (2) and (4) can be reduced by more sophisticated wikification and gloss classification, respectively. An example of error (3) is that Wikipedia entry Spanish_flu is empty because it is redirected

to 1918_flu_pandemic. A simple remedy of partial help to error (3) is to resolve such empty concepts using Wikipedia redirect relations.

Analysis of Event Detection. Two major error sources of the DS-BLSTM model are nouns and phrases. Our training data generated from SemCor is 11 times larger than SW100 with respect to the number of event nuggets. Still, many nouns and phrases do not appear in the training data, making correct predictions difficult. As shown in Table 5(a), the most difficult domain is 'Disease' where numerous domain-specific terms, such as migraine and bubonic plague, can appear even in simplified text of Simple Wikipedia, but not in SemCor at all.

7 Related Work

Most prior work has addressed event detection using supervised models based on symbolic features. Some studies employ token-level classifiers (Ahn, 2006; Ji and Grishman, 2008; Liao and Grishman, 2010b; Hong et al., 2011; Berant et al., 2014), while others cast event detection as a sequence labeling problem and apply structured prediction models (McClosky et al., 2011; Lu and Roth, 2012; Li et al., 2013; Li et al., 2014; Araki and Mitamura, 2015; Yang and Mitchell, 2016). Recent work has explored neural network models with distributional features (Ghaeini et al., 2016; Nguyen et al., 2016; Chen et al., 2015; Nguyen and Grishman, 2015; Feng et al., 2016). These models are typically trained on a small amount of human-annotated data in closed domains and thus subject to overfitting.

Semi-supervised and unsupervised approaches are less studied than supervised ones in event detection. Several studies leverage bootstrapping methods to find new patterns for similar events (Liao and Grishman, 2010a; Liu and Strzalkowski, 2012; Huang and Riloff, 2012; Huang and Riloff, 2013). Others explore self-training (Liao and Grishman, 2011), event vector representation (Peng et al., 2016), tensor-based composition (Huang et al., 2016), and distant supervision (Chen et al., 2017). However, their models focus on predicate-argument structures and are validated in a few domains, mostly in ACE. It is unclear how well they scale to the open domain, particularly to phrases and proper nouns.

Several lines of recent work refine the definition of events. Rich ERE (LDC, 2015) defines events under a particular event ontology mainly from the syntactic perspective, without clarifying what semantically constitutes events. The ISO-TimeML specification (Pustejovsky et al., 2010) defines events as "something that can be said to obtain or hold true, to happen or to occur." This definition is consistent with Bach's eventualities and the closest to ours. The studies described above have narrower coverage of events than our work in terms of domains, syntactic types, or multi-word expressions.

8 Conclusion and Future Work

We have introduced open-domain event detection, a new event detection paradigm whose goal is to detect all kinds of events regardless of domains. Due to the ubiquity and ambiguities of events, human annotation of events in the open domain is substantially expensive. We presented a distant supervision method that is able to generate high-quality training data automatically, obviating the need for human annotation. Our distantly supervised model is not bounded to any particular datasets and offers a versatile solution for event detection. Our experiment shows that the model outperforms supervised models in both in-domain and out-domain settings.

There are numerous avenues for future work. One could conduct experiments on normal English text, such as newspaper articles, in various domains. We plan on event coreference resolution to detect eventive pronouns and demonstrative determiners. We recognize that event types and epistemic status are key features for event coreference resolution, and thus extracting them is an important problem.

Acknowledgements

This publication was partly made possible by grant NPRP-08-1337-1-243 from the Qatar National Research Fund (a member of the Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

David Ahn. 2006. The stages of event extraction. In *Proceedings of COLING/ACL Workshop on Annotating and Reasoning about Time and Events*, pages 1–8.

Jun Araki and Teruko Mitamura. 2015. Joint event trigger identification and event coreference resolution with structured perceptron. In *Proceedings of EMNLP*, pages 2074–2080.

Emmon Bach. 1986. The algebra of events. *Linguistics and Philosophy*, 9:5–16.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of COLING*, pages 86–90.

Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, Peter Clark, and Christopher D. Manning. 2014. Modeling biological processes for reading comprehension. In *Proceedings of EMNLP*, pages 1499–1510.

Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of ACL/IJCNLP*, pages 167–176.

Yubo Chen, Kang Liu, and Jun Zhao. 2017. Automatically labeled data generation for large scale event extraction. In *Proceedings of ACL*.

Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? Lexical diversity and event coreference resolution. In *Proceedings of LREC*, pages 4545–4552.

Tom De Smedt and Walter Daelemans. 2012. Pattern for python. *Journal of Machine Learning Research*, 13:2063–2067.

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 LREC*, pages 837–840.

Xiaocheng Feng, Lifu Huang, Duyu Tang, Heng Ji, Bing Qin, and Ting Liu. 2016. A language-independent neural network for event detection. In *Proceedings of ACL*, pages 66–71.

Reza Ghaeini, Xiaoli Fern, Liang Huang, and Prasad Tadepalli. 2016. Event nugget detection with forward-backward recurrent neural networks. In *Proceedings of ACL*, pages 369–373.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Ralph Grishman and Beth Sundheim. 1996. Message understanding conference-6: A brief history. In *Proceedings of COLING*, pages 466–471.

Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *Proceedings of ACL-HLT*, pages 1127–1136.

Ruihong Huang and Ellen Riloff. 2012. Bootstrapped training of event extraction classifiers. In *Proceedings of EACL*, pages 286–295.

Ruihong Huang and Ellen Riloff. 2013. Multi-faceted event recognition with bootstrapped dictionaries. In *Proceedings of NAACL-HLT*, pages 41–51.

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 ACL*, pages 258–268.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of ACL/IJCNLP*, pages 1681–1691.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL-HLT*, pages 254–262.

Jin-Dong Kim, Tomoko Ohta, and Jun'ichi Tsujii. 2008. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 9:10.

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 BioNLP-ST Workshop*, pages 1–9.

LDC, 2015. *DEFT Rich ERE Annotation Guidelines: Events*. Linguistic Data Consortium. Version 3.0.

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint entity and event coreference resolution across documents. In *Proceedings of EMNLP/CoNLL*, pages 489–500.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of EMNLP*, pages 188–197.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of ACL*, pages 73–82.

Qi Li, Heng Ji, Yu Hong, and Sujian Li. 2014. Constructing information networks using one single model. In *Proceedings of EMNLP*, pages 1846–1851.

Shasha Liao and Ralph Grishman. 2010a. Filtered ranking for bootstrapping in event extraction. In *Proceedings of COLING*, pages 680–688.

Shasha Liao and Ralph Grishman. 2010b. Using document level cross-event inference to improve event extraction. In *Proceedings of ACL*, pages 789–797.

Shasha Liao and Ralph Grishman. 2011. Can document selection help semi-supervised learning? A case study on event extraction. In *Proceedings of ACL-HLT*, pages 260–265.

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *Proceedings of ICLR*.

Ting Liu and Tomek Strzalkowski. 2012. Bootstrapping events and relations from text. In *Proceedings of EACL*, pages 296–305.

Wei Lu and Dan Roth. 2012. Automatic event extraction with structured preference modeling. In *Proceedings of ACL*, pages 835–844.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings ACL: System Demonstrations*, pages 55–60.

David McClosky, Mihai Surdeanu, and Christopher Manning. 2011. Event extraction as dependency parsing. In *Proceedings of ACL-HLT*, pages 1626–1635.

Alejandro Metke-Jimenez and Sarvnaz Karimi. 2016. Concept identification and normalisation for adverse drug event discovery in medical forums. In *Proceedings of the Workshop on Biomedical Data Integration and Discovery*.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. 1990. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Proceedings of the Workshop on Human Language Technology*, pages 303–308.

Teruko Mitamura, Yukari Yamakawa, Susan Holm, Zhiyi Song, Ann Bies, Seth Kulick, and Stephanie Strassel. 2015. Event nugget annotation: Processes and issues. In *Proceedings of NAACL-HLT 2015 Workshop on Events: Definition, Detection, Coreference, and Representation*, pages 66–76.

Teruko Mitamura, Zhengzhong Liu, and Eduard Hovy. 2016. Overview of TAC-KBP 2016 Event Nugget track. In *Proceedings of Text Analysis Conference*.

Marc Moens and Mark Steedman. 1988. Temporal ontology and temporal reference. *Computational Linguistics*, 14(2):15–28.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of ACL/IJCNLP*, pages 365–371.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of NAACL-HLT*, pages 300–309.

Tim O'Gorman, Kristin Wright-Bettner, and Martha Palmer. 2016. Richer Event Description: Integrating event coreference with temporal, causal and bridging annotation. In *Proceedings of the 2nd Workshop on Computing News Storylines*, pages 47–56.

Boyan Onyshkevych, Mary Ellen Okurowski, and Lynn Carlson. 1993. Tasks, domains, and languages for information extraction. In *Proceedings of the TIPSTER Text Program: Phase I*, pages 123–133.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–105.

Martha Palmer, Will Styler, Kevin Crooks, and Tim O'Gorman, 2016. *Richer Event Description (RED) Annotation Guidelines*. University of Colorado at Boulder. Version 1.7, `https://github.com/timjogorman/RicherEventDescription/blob/master/guidelines.md`.

Haoruo Peng, Yangqiu Song, and Dan Roth. 2016. Event detection and co-reference with minimal supervision. In *Proceedings of EMNLP*, pages 392–402.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543.

Sameer S. Pradhan, Lance Ramshaw, Ralph Weischedel, Jessica MacBride, and Linnea Micciulla. 2007. Unrestricted coreference: Identifying entities and events in OntoNotes. In *Proceedings of the 2007 International Conference on Semantic Computing*, pages 446–453.

Stephen G. Pulman. 1997. Aspectual shift as type coercion. *Transactions of the Philological Society*, 95(2):279–317.

James Pustejovsky, José M. Castaño, Robert Ingria, Roser Sauri, Robert J. Gaizauskas, Andrea Setzer, and Graham Katz. 2003. TimeML: Robust specification of event and temporal expressions in text. In *Fifth International Workshop on Computational Semantics (IWCS-5)*, pages 28–34.

James Pustejovsky, Kiyong Lee, Harry Bunt, and Laurent Romary. 2010. ISO-TimeML: An international standard for semantic annotation. In *Proceedings of LREC*, pages 394–397.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of ACL*, pages 1375–1384.

Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. 2012. Open domain event extraction from Twitter. In *Proceedings of SIGKDD*, pages 1104–1112.

Michael Roth and Mirella Lapata. 2016. Neural semantic role labeling with dependency path embeddings. In *Proceedings of ACL*, pages 1192–1202.

Roser Saurí, Robert Knippen, Marc Verhagen, and James Pustejovsky. 2005. Evita: A robust event recognizer for QA systems. In *Proceedings of HLT/EMNLP*, pages 700–707.

Nathan Schneider, Emily Danchik, Chris Dyer, and Noah A. Smith. 2014. Discriminative lexical semantic segmentation with gaps: Running the MWE gamut. *Transactions of the Association for Computational Linguistics*, 2:193–206.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of ACL*, pages 384–394.

Tuangthong Wattarujeekrit, Parantu K. Shah, and Nigel Collier. 2004. PASBio: predicate-argument structures for event extraction in molecular biology. *BMC Bioinformatics*, 5:155.

Ralph Weischedel, Eduard Hovy, Mitchell Marcus, Martha Palmer, Robert Belvin, Sameer Pradhan, Lance Ramshaw, and Nianwen Xue. 2011. OntoNotes: A large training corpus for enhanced processing. In *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*, pages 54–63. Springer.

Bishan Yang and Tom Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of NAACL-HLT*, pages 289–299.

Wenpeng Yin and Hinrich Schütze. 2015. Discriminative phrase embedding for paraphrase identification. In *Proceedings of NAACL-HLT*, pages 1368–1373.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of ACL: System Demonstrations*, pages 78–83.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of ACL*, pages 207–212.