Event Detection and Co-reference with Minimal Supervision

Haoruo Peng[1] and **Yangqiu Song**[2] and **Dan Roth**[1]
[1]University of Illinois, Urbana-Champaign
[2]Department of Computer Science and Engineering,
Hong Kong University of Science and Technology
[1]{hpeng7,danr}@illinois.edu, [2]yqsong@cse.ust.hk

Abstract

An important aspect of natural language understanding involves recognizing and categorizing events and the relations among them. However, these tasks are quite subtle and annotating training data for machine learning based approaches is an expensive task, resulting in supervised systems that attempt to learn complex models from small amounts of data, which they over-fit. This paper addresses this challenge by developing an event detection and co-reference system with minimal supervision, in the form of a few event examples. We view these tasks as semantic similarity problems between event mentions or event mentions and an ontology of types, thus facilitating the use of large amounts of out of domain text data. Notably, our semantic relatedness function exploits the structure of the text by making use of a semantic-role-labeling based representation of an event.

We show that our approach to event detection is competitive with the top supervised methods. More significantly, we outperform state-of-the-art supervised methods for event co-reference on benchmark data sets, and support significantly better transfer across domains.

1 Introduction

Natural language understanding involves, as a key component, the need to understand events mentioned in texts. This entails recognizing elements such as agents, patients, actions, location and time, among others. Understanding events also necessitates understanding relations among them and, as a minimum, determining whether two snippets of text represent the same event or not – the event co-reference problem. Events have been studied for years, but they still remain a key challenge. One reason is that the frame-based structure of events necessitates addressing multiple coupled problems that are not easy to study in isolation. Perhaps an even more fundamental difficulty is that it is not clear whether our current set of events' definitions is adequate (Hovy et al., 2013). Thus, given the complexity and fundamental difficulties, the current evaluation methodology in this area focuses on a limited domain of events, e.g. 33 types in ACE 2005 (NIST, 2005) and 38 types in TAC KBP (Mitamura et al., 2015). Consequently, this allows researchers to train supervised systems that are tailored to these sets of events and that overfit the small domain covered in the annotated data, rather than address the realistic problem of understanding events in text.

In this paper, we pursue an approach to understanding events that we believe to be more feasible and scalable. Fundamentally, event detection is about identifying whether an event in context is semantically related to a set of events of a specific type; and, event co-reference is about whether two event mentions are semantically similar enough to indicate that the author intends to refer to the same thing. Therefore, if we formulate event detection and co-reference as semantic relatedness problems, we can scale it to deal with a lot more types and, potentially, generalize across domains. Moreover, by doing so, we facilitate the use of a lot of data that is not part of the existing annotated event collections and not even from the same domain. The key chal-

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 392–402,
Austin, Texas, November 1-5, 2016. ©2016 Association for Computational Linguistics

	Supervised	Unsupervised	MSEP
Guideline	✓	✓	✓
In-domain Data	✓	✓	✗
Data Annotation	✓	✗	✗

Table 1: **Comparing requirements of MSEP and other methods.** Supervised methods need all three resources while MSEP only needs an annotation guideline (as event examples).

lenges we need to address are those of how to represent events, and how to model event similarity; both are difficult partly since events have *structure*.

We present a general event detection and coreference framework, which essentially requires no labeled data. In practice, in order to map an event mention to an event ontology, as a way to communicate with a user, we just need a few event examples, in plain text, for each type a user wants to extract. This is a reasonable setting; after all, giving examples is the easiest way of defining event types, and is also how information needs are defined to annotators - by providing examples in the annotation guideline.[1] Our approach makes less assumptions than standard *unsupervised* methods, which typically require a *collection* of instances and exploit similarities among them to eventually learn a model. Here, given event type definitions (in the form of a few examples), we can classify a *single* event into a provided ontology and determine whether two events are co-referent. In this sense, our approach is similar to what has been called *dataless classification* (Chang et al., 2008; Song and Roth, 2014). Table 1 summarizes the difference between our approach, **MSEP** (Minimally Supervised Event Pipeline)[2], and other methods.

Our approach builds on two key ideas. First, to represent event structures, we use the general purpose nominal and verbial semantic role labeling (SRL) representation. This allows us to develop a structured representation of an event. Second, we embed event components, while maintaining the structure, into multiple semantic spaces, in-

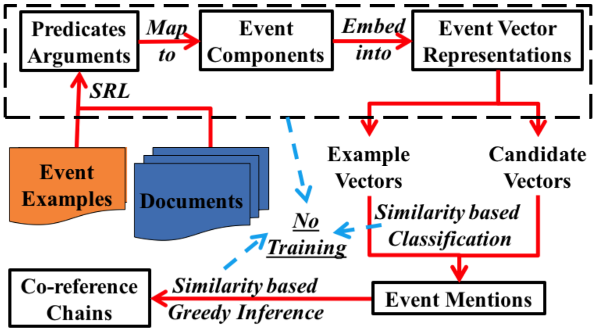

Figure 1: An overview of the end-to-end MSEP system. “Event Examples” are the only supervision here, which produce “Example Vectors”. No training is needed for MSEP.

duced at a contextual, topical, and syntactic levels. These semantic representations are induced from large amounts of text in a way that is completely independent of the tasks at hand, and are used to represent both event mentions and event types into which we classify our events. The combination of these semantic spaces, along with the structured vector representation of an event, allow us to directly determine whether a candidate event mention is a valid event or not and, if it is, of which type. Moreover, with the same representation, we can evaluate event similarities and decide whether two event mentions are co-referent. Consequently, the proposed MSEP, can also adapt to new domains without any training.

An overview of the system is shown in Figure 1. A few event examples are *all* the supervision MSEP needs; even the few decision thresholds needed to be set are determined on these examples, once and for all, and are used for *all* test cases we evaluate on. We use two benchmark datasets to compare MSEP with baselines and supervised systems. We show that MSEP performs favorably relative to state-of-the-art supervised systems; the co-reference module, in fact, outperforms supervised approaches on B^3 and CEAF metrics. The superiority of MSEP is also demonstrated in across domain settings.

2 The MSEP System

2.1 Structured Vector Representation

There is a parallel between event structures and sentence structures. Event triggers are mostly predicates of sentences or clauses. Predicates can be sense disambiguated, which roughly corresponds to

[1] Event examples also serve for disambiguation purposes. For example, using “U.S. forces bombed Baghdad.” to exemplify an *attack* type, disambiguates it from a *heart attack*.

[2] Available at http://cogcomp.cs.illinois.edu/page/download_view/eventPipeline .

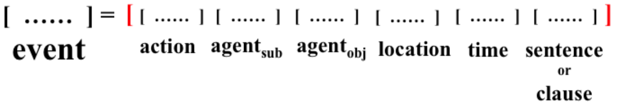

Figure 2: **Basic event vector representation.** Event vector is the concatenation of vectors corresponding to action, agent$_{sub}$, agent$_{obj}$, location, time and sentence/clause.

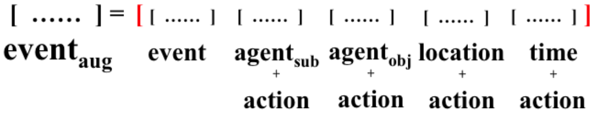

Figure 3: **Augmented event vector representation.** Event vector is the concatenation of vectors corresponding to basic event vector representation, agent$_{sub}$ + action, agent$_{obj}$ + action, location + action and time + action. Here, "+" means that we first put text fragments together and then convert the combined text fragment into an ESA vector.

event types. Event arguments are largely entity mentions or temporal/spatial arguments. They serve as specific roles in events, similarly to SRL arguments that are assigned role labels for predicates.

We use the Illinois SRL (Punyakanok et al., 2004) tool to pre-process the text. We evaluate the SRL coverage on both event triggers and event arguments, shown in Table 2.[3] For event triggers, we only focus on recall since we expect the event mention detection module to filter out most non-trigger predicates. Results show a good coverage of SRL predicates and arguments on event triggers and arguments. Even though we only get approximate event arguments, it is easier and more reliable to categorize them into five abstract roles, than to determine the exact role label with respect to event triggers.

ACE		Precision	Recall	F1
Predicates over Triggers	Verb-SRL	—	93.2	—
	Nom-SRL	—	87.5	—
	All	—	91.9	—
SRL Args over Event Args	Verb-SRL	90.4	85.7	88.0
	Nom-SRL	92.5	73.5	81.9
	All	90.9	82.3	86.4
TAC KBP		Precision	Recall	F1
Predicates over Triggers	Verb-SRL	—	90.6	—
	Nom-SRL	—	85.5	—
	All	—	88.1	—
SRL Args over Event Args	Verb-SRL	89.8	83.6	86.6
	Nom-SRL	88.2	69.9	78.0
	All	89.5	81.0	85.0

Table 2: **Semantic role labeling coverage.** We evaluate both "Predicates over Triggers" and "SRL Arguments over Event Arguments". "All" stands for the combination of Verb-SRL and Nom-SRL. The evaluation is done on all data.

We identify the five most important and abstract event semantic components: action, agent$_{sub}$, agent$_{obj}$, location and time. To map SRL arguments to these event arguments, we run through the following procedures: 1) set predicates as actions, and preserve SRL negations for actions, 2) set SRL subject as agent$_{sub}$, 3) set SRL object and indirect object as agent$_{obj}$, 4) set SRL spatial argument as event location. If there is no such SRL label, we then scan for any NER location label within the sentence/clause to which the action belongs. We set the location according to NER information if it exists. 5) We set the SRL temporal argument as event time. If there is no such SRL label, we then use the Illinois Temporal Expression Extractor (Zhao et al., 2012) to find the temporal argument within an event's sentence/clause. 6) We allow one or more missing event arguments among agent$_{sub}$, agent$_{obj}$, location or time, but require actions to always exist.

Given the above structured information, we convert each event component to its corresponding vector representation, discussed in detail in Section 3. We then concatenate the vectors of all components together in a specific order: action, agent$_{sub}$, agent$_{obj}$, location, time and sentence/clause. We treat the whole sentence/clause, to which the "ac-

[3] We place events in two categories, verb or noun, according to the part-of-speech tag of the trigger. We evaluate verb-SRL on events with verb triggers, nom-SRL on events with noun triggers, and the overall performance on all events. When evaluating, we allow partial overlaps.

tion" belongs, as context, and we append its corresponding vector to the event representation. This basic event vector representation is illustrated in Fig. 2. If there are missing event arguments, we set the corresponding vector to be "NIL" (we set each position as "NaN"). We also augment the event vector representation by concatenating more text fragments to enhance the interactions between the action and other arguments, as shown in Fig. 3. Essentially, we flatten the event structure to preserve the alignment of event arguments so that the structured information can be reflected in our vector space.

2.2 Event Mention Detection

Motivated by the seed-based event trigger labeling technique employed in Bronstein et al. (2015), we turn to ACE annotation guidelines for event examples described under each event type label. For instance, the ACE-2005 guidelines list the example "Mary Smith joined Foo Corp. in June 1998." for label "START-POSITION". Altogether, we collect 172 event examples from 33 event types (5 each on average).[4] We can then get vector representations for these example events following the procedures in Sec. 2.1. We define the *event type representation* as the numerical average of all vector representations corresponding to example events under that type. We use the similarity between an event candidate with the event type representation to determine whether the candidate belongs to an event type:

$$\begin{aligned} S(e_1, e_2) &= \frac{vec(e_1) \cdot vec(e_2)}{\|vec(e_1)\| \cdot \|vec(e_2)\|} \\ &= \frac{\sum_a vec(a_1) \cdot vec(a_2)}{\sqrt{\sum_a \|vec(a_1)\|^2} \cdot \sqrt{\sum_a \|vec(a_2)\|^2}}, \end{aligned} \tag{1}$$

where e_1 is the candidate, e_2 the type ($vec(e_2)$ is computed as average of event examples), a_1, a_2 are components of e_1, e_2 respectively. We use the notation $vec(\cdot)$ for corresponding vectors. Note that there may be missing event arguments (NIL). In such cases, we use the average of all non-NIL similarity scores for that particular component as the contributed score. Formally, we define $S_{pair}(a = \text{NIL})$ and $S_{single}(a = \text{NIL})$ as follows:

$$\begin{aligned} S_{pair}(a = \text{NIL}) &= vec(\text{NIL}) \cdot vec(a_2) \\ &= vec(a_1) \cdot vec(\text{NIL}) \\ &= \sum_{a_1, a_2 \neq \text{NIL}} \frac{vec(a_1) \cdot vec(a_2)}{\#|a_1, a_2 \neq \text{NIL}|}, \end{aligned}$$

$$S_{single}(a = \text{NIL}) = \sqrt{\frac{\sum_{a \neq \text{NIL}} \|vec(a)\|^2}{\#|a \neq \text{NIL}|}}.$$

Thus, when we encounter missing event arguments, we use $S_{pair}(a = \text{NIL})$ to replace the corresponding term in the numerator in $S(e_1, e_2)$ while using $S_{single}(a = \text{NIL})$ in the denominator. These average contributed scores are corpus independent, and can be pre-computed ahead of time. We use a cut-off threshold to determine that an event does not belong to any event types, and can thus be eliminated. This threshold is set by tuning only on the set of event examples, which is corpus independent.[5]

2.3 Event Co-reference

Similar to the mention-pair model in entity co-reference (Ng and Cardie, 2002; Bengtson and Roth, 2008; Stoyanov et al., 2010), we use cosine similarities computed from pairs of event mentions: $S(e_1, e_2)$ (as in Eq. (1)).

Before applying the co-reference model, we first use external knowledge bases to identify conflict events. We use the Illinois Wikification (Cheng and Roth, 2013) tool to link event arguments to Wikipedia pages. Using the Wikipedia IDs, we map event arguments to Freebase entries. We view the top-level Freebase type as the event argument type. An event argument can contain multiple wikified entities, leading to multiple Wikipedia pages and thus a set of Freebase types. We also augment the argument type set with NER labels: PER (person) and ORG (organization). We add either of the NER labels if we detect such a named entity.

For each pair of events, we check event arguments agent$_{sub}$ and agent$_{obj}$ respectively. If none of the types for the aligned event arguments match, this pair is determined to be in conflict. If the event argument is missing, we deem it compatible with any type. In this procedure, we generate a set of event pairs Set_{conflict} that will not get co-reference links.

[4] See supplementary materials for the full list of examples.

[5] See Sec. 4.4 for details.

Given the event mention similarity as well as the conflicts, we perform event co-reference inference via a left-linking greedy algorithm, i.e. co-reference decisions are made on each event from left to right, one at a time. Without loss of generality, for event $e_{k+1}, \forall k \geq 1$, we first choose a linkable event to its left with the highest event-pair similarity:

$$e_p = \arg \max_{\substack{e \in \{e_1, e_2, \ldots, e_k\} \\ e \notin Set_{\text{conflict}}}} S(e, e_{k+1}).$$

We make co-reference links when $S(e_p, e_{k+1})$ is higher than a cut-off threshold, which is also tuned only on event examples ahead of time. Otherwise, event e_{k+1} is not similar enough to any of its antecedents, and we make it the start of a new cluster.

3 Vector Representations

We experiment with different methods to convert event components into vector representations. Specifically, we use Explicit Semantic Analysis (ESA), Brown Cluster (BC), Word2Vec (W2V) and Dependency-Based Word Embedding (DEP) respectively to convert text into vectors. We then concatenate all components of an event together to form a structured vector representation.

Explicit Semantic Analysis ESA uses Wikipedia as an external knowledge base to generate concepts for a given fragment of text (Gabrilovich and Markovitch, 2009). ESA first represents a given text fragment as a TF-IDF vector, then uses an inverted index for each word to search the Wikipedia corpus. The text fragment representation is thus a weighted combination of the concept vectors corresponding to its words. We use the same setting as in Chang et al. (2008) to filter out pages with fewer than 100 words and those containing fewer than 5 hyperlinks. To balance between the effectiveness of ESA representations and its cost, we use the 200 concepts with the highest weights. Thus, we convert each text fragment to a very sparse vector of millions of dimensions (but we just store 200 non-zero values).

Brown Cluster BC was proposed by Brown et al. (1992) as a way to support abstraction in NLP tasks, measuring words' distributional similarities. This method generates a hierarchical tree of word clusters by evaluating the word co-occurrence based on a n-gram model. Then, paths traced from root to leaves can be used as word representations. We use the implementation by Song and Roth (2014), generated over the latest Wikipedia dump. We set the maximum tree depth to 20, and use a combination of path prefixes of length 4,6 and 10 as our BC representation. Thus, we convert each word to a vector of $2^4 + 2^6 + 2^{10} = 1104$ dimensions.

Word2Vec We use the skip-gram tool by Mikolov et al. (2013) over the latest Wikipedia dump, resulting in word vectors of dimensionality 200.

Dependency-Based Embedding DEP is the generalization of the skip-gram model with negative sampling to include arbitrary contexts. In particular, it deals with dependency-based contexts, and produces markedly different embeddings. DEP exhibits more functional similarity than the original skip-gram embeddings (Levy and Goldberg, 2014). We directly use the released 300-dimension word embeddings[6].

Note that it is straightforward text-vector conversion for ESA. But for BC, W2V and DEP, we first remove stop words from the text and then average, element-wise, all remaining word vectors to produce the resulting vector representation of the text fragment.

4 Experiments

4.1 Datasets

ACE The ACE-2005 English corpus (NIST, 2005) contains fine-grained event annotations, including event trigger, argument, entity, and time-stamp annotations. We select 40 documents from newswire articles for event detection evaluation and the rest for training (same as Chen et al. (2015)). We do 10-fold cross-validation for event co-reference.

TAC-KBP The TAC-KBP-2015 corpus is annotated with event nuggets that fall into 38 types and co-reference relations between events. [7] We use the train/test data split provided by the official TAC-

[6] https://levyomer.wordpress.com/2014/04/25/dependency-based-word-embeddings

[7] The event ontology of TAC-KBP (based on ERE annotation) is almost the same to that of ACE. To adapt our system to the TAC-KBP corpus, we use all ACE event seeds of "Contact.Phone-Write" for "Contact.Correspondence" and separate ACE event seeds of "Movement.Transport" into "Movement.TransportPerson" and "Movement.TransportArtifact" by manual checking. So, we use exactly the same set of event seeds for TAC-KBP with only these two changes.

	#Doc	#Sent.	#Men.	#Cluster
ACE(All)	599	15,494	5,268	4,046
ACE(Test)	40	672	289	222
TAC-KBP(All)	360	15,824	12,976	7,415
TAC-KBP(Test)	202	8,851	6,438	3,779

Table 3: **Statistics for the ACE and TAC-KBP corpora.** #Sent. is the number of sentences, #Men. is the number of event mentions, and #Cluster is the number of event clusters (including singletons). Note that the proposed MSEP does not need any training data.

2015 Event Nugget Evaluation Task.

Statistics for the ACE and TAC-KBP corpora is shown in Table 3. Note that the training set and cross-validation is only for competing supervised methods. For MSEP, we only need to run on each corpus once for testing.

4.2 Compared Systems

For event detection, we compare with **DM-CNN** (Chen et al., 2015), the state-of-art supervised event detection system. We also implement another supervised model, named *supervised structured event detection* **SSED** system following the work of Sammons et al. (2015). The system utilizes rich semantic features and applies a trigger identification classifier on every SRL predicate to determine the event type. For event co-reference, **Joint** (Chen et al., 2009) is an early work based on supervised learning. We also report **HDP-Coref** results as an unsupervised baseline (Bejan and Harabagiu, 2010), which utilizes nonparametric Bayesian models. Moreover, we create another unsupervised event co-reference baseline (**Type+SharedMen**): we treat events of the same type which share at least one co-referent entity (inside event arguments) as co-referred. On TAC-KBP corpus, we report results from the top ranking system of the TAC-2015 Event Nugget Evaluation Task as **TAC-TOP**.

We name our event mention detection module in MSEP *similarity-based event mention detection* **MSEP-EMD** system. For event co-reference, the proposed similarity based co-reference detection **MSEP-Coref** method has a number of variations depending on the modular text-vector conversion method (**ESA, BC, W2V, DEP**), whether we use augmented ESA vector representation (**AUG**)[8], and whether we use knowledge during co-reference inference (**KNOW**). We also develop a supervised event co-reference system following the work of Sammons et al. (2015), namely **Supervised**$_{\textbf{Base}}$. We also add additional event vector representations[9] as features to this supervised system and get **Supervised**$_{\textbf{Extend}}$.

4.3 Evaluation Metrics

For event detection, we use standard precision, recall and F1 metrics. For event co-reference, we compare all systems using standard F1 metrics: MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), Entity-based CEAF (CEAF$_e$) (Luo, 2005) and BLANC (Recasens and Hovy, 2011). We use the average scores (AVG) of these four metrics as the main comparison metric.[10]

4.4 Results for Event detection

The performance comparison for event detection is presented in Table 4. On both ACE and TAC-KBP, parameters of SSED are tuned on a development set (20% of randomly sampled training documents). The cut-off threshold for MSEP-EMD is tuned on the 172 event examples ahead of time by optimizing the F1 score on the event seed examples. Note that different text-vector conversion methods lead to different cut-off thresholds, but they remain fixed for all the test corpus. Results show that SSED achieves state-of-the-art performance. Though MSEP-EMD's performance is below the best supervised system, it is very competitive. Note that both SSED and MSEP-EMD use SRL predicates as input and thus can further improve with a better SRL module.

4.5 Results for Event Co-reference

The performance of different systems for event co-reference based on gold event triggers is shown in Table 5. The co-reference cut-off threshold is tuned by optimizing the CoNLL average score on ten se-

[8] It is only designed for ESA because the ESA vector for two concatenated text fragments is different from the sum of the ESA vectors of individual text fragments, unlike other methods.

[9] We add the best event vector representation empirically.

[10] We use the latest scorer (v1.7) provided by TAC-2015 Event Nugget Evaluation for all metrics.

ACE (Test Data)		Precision	Recall	F1
Span	DMCNN	80.4	67.7	73.5
	SSED	76.6	71.5	**74.0**
	MSEP-EMD	75.6	69.8	72.6
Span+Type	DMCNN	75.6	63.6	**69.1**
	SSED	71.3	66.5	68.8
	MSEP-EMD	70.4	65.0	67.6
TAC-KBP (Test Data)		Precision	Recall	F1
Span	SSED	77.2	55.9	64.8
	TAC-TOP	—	—	**65.3**
	MSEP-EMD	76.5	54.5	63.5
Span+Type	SSED	69.9	48.8	57.5
	TAC-TOP	—	—	**58.4**
	MSEP-EMD	69.2	47.8	56.6

Table 4: **Event detection (trigger identification) results.** "Span"/"Type" means span/type match respectively.

lected ACE documents. The threshold is then fixed, thus we do not change it when evaluating on the TAC-KBP corpus. As we do cross-validation on ACE, we exclude these ten documents from test at all times.[11] Results show that the proposed MSEP event co-reference system significantly outperforms baselines and achieves the same level of performance of supervised methods (82.9 v.s. 83.3 on ACE and 73.8 v.s. 74.4 on TAC-KBP). MSEP achieves better results on B^3 and CEAF$_e$ than supervised methods. Note that supervised methods usually generate millions of features (2.5M on ACE and 1.8M on TAC-KBP for Supervised$_{\text{Base}}$). In contrast, MSEP only has several thousands of non-zero dimensions in event representations. This means that our structured vector representations, through derived without explicit annotations, are far more expressive than traditional features. When we add the event vector representation (augmented ESA) as features in Supervised$_{\text{Extend}}$, we improve the overall performance by more than 1 point. When tested individually, DEP performs the best among the four text-vector conversion methods while BC performs the worst. A likely reason is that BC has too few dimensions while DEP constructs the longest vector. However, the results show that our augmented ESA representation (Fig. 2) achieves even better results.

When we use knowledge to detect conflicting events during inference, the system further improves. Note that event arguments for the proposed MSEP are predicted by SRL. We show that replacing them with gold event arguments, only slightly improves the overall performance, indicating that SRL arguments are robust enough for the event co-reference task.

4.6 End-to-End Event Co-reference Results

Table 6 shows the performance comparison for end-to-end event co-reference. We use both SSED and MSEP-EMD as event detection modules and we evaluate on standard co-reference metrics. Results on TAC-KBP show that "SSED+Supervised$_{\text{Extend}}$" achieves similar performance to the TAC top ranking system while the proposed MSEP event co-reference module helps to outperform supervised methods on B^3 and CEAF$_e$ metrics.

4.7 Domain Transfer Evaluation

To demonstrate the superiority of the adaptation capabilities of the proposed MSEP system, we test its performance on new domains and compare with the supervised system. TAC-KBP corpus contains two genres: newswire (NW) and discussion forum (DF), and they have roughly equal number of documents. When trained on NW and tested on DF, supervised methods encounter out-of-domain situations. However, the MSEP system can adapt well.[12] Table 7 shows that MSEP outperforms supervised methods in out-of-domain situations for both tasks. The differences are statistically significant with $p < 0.05$.

5 Related Work

Event detection has been studied mainly in the newswire domain as the task of detecting event triggers and determining event types and arguments. Most earlier work has taken a pipeline approach where local classifiers identify triggers first, and then arguments (Ji and Grishman, 2008; Liao and

[11] We regard this tuning procedure as "independent" and "ahead of time" because of the following reasons: 1) We could have used as threshold-tuning co-reference examples a few news documents from other sources; we just use ACE documents as a data source for simplicity. 2) We believe that the threshold only depends on event representation (the model) rather than data. 3) Tuning a single decision threshold is much cheaper than tuning a whole set of model parameters.

[12] Note that the supervised method needs to be re-trained and its parameters re-tuned while MSEP does not need training and its cut-off threshold is fixed ahead of time using event examples.

ACE (Cross-Validation)		MUC	B^3	CEAF_e	BLANC	AVG
Supervised	Graph	—	—	84.5	—	—
	Joint	74.8	92.2	87.0	—	—
	Supervised$_{\text{Base}}$	73.6	91.6	85.9	82.2	83.3
	Supervised$_{\text{Extend}}$	**74.9**	92.8	87.1	**83.8**	**84.7**
Unsupervised	Type+SharedMen	59.1	83.2	76.0	72.9	72.8
	HDP-Coref	—	83.8	76.7	—	—
MSEP	MSEP-Coref$_{\text{ESA}}$	65.9	91.5	85.3	81.8	81.1
	MSEP-Coref$_{\text{BC}}$	65.0	89.8	83.7	80.9	79.9
	MSEP-Coref$_{\text{W2V}}$	65.1	90.1	83.6	81.5	80.1
	MSEP-Coref$_{\text{DEP}}$	65.9	92.3	85.6	81.5	**81.3**
	MSEP-Coref$_{\text{ESA+AUG}}$	67.4	92.6	86.0	82.6	**82.2**
	MSEP-Coref$_{\text{ESA+AUG+KNOW}}$	68.0	**92.9**	**87.4**	83.2	**82.9**
	MSEP-Coref$_{\text{ESA+AUG+KNOW}}$ (GA)	68.8	92.5	87.7	83.4	83.1
TAC-KBP (Test Data)		MUC	B^3	CEAF_e	BLANC	AVG
Supervised	TAC-TOP	—	—	—	—	**75.7**
	Supervised$_{\text{Base}}$	63.8	83.8	75.8	74.0	74.4
	Supervised$_{\text{Extend}}$	**65.3**	84.7	76.8	**75.1**	**75.5**
Unsupervised	Type+SharedMen	56.4	77.5	69.6	68.7	68.1
MSEP	MSEP-Coref$_{\text{ESA}}$	57.7	83.9	76.9	72.9	72.9
	MSEP-Coref$_{\text{BC}}$	56.9	81.8	76.2	71.7	71.7
	MSEP-Coref$_{\text{W2V}}$	57.2	82.1	75.9	72.3	71.9
	MSEP-Coref$_{\text{DEP}}$	58.2	83.3	76.7	72.8	**72.8**
	MSEP-Coref$_{\text{ESA+AUG}}$	59.0	84.5	77.3	72.5	**73.3**
	MSEP-Coref$_{\text{ESA+AUG+KNOW}}$	59.9	**84.9**	**77.3**	73.1	**73.8**
	MSEP-Coref$_{\text{ESA+AUG+KNOW}}$ (GA)	60.5	84.0	77.7	73.5	73.9

Table 5: **Event Co-reference Results on Gold Event Triggers.** "MSEP-Coref$_{\text{ESA,BC,W2V,DEP}}$" are variations of the proposed MSEP event co-reference system using ESA, Brown Cluster, Word2Vec and Dependency Embedding representations respectively. "MSEP-Coref$_{\text{ESA+AUG}}$" uses augmented ESA event vector representation and "MSEP-Coref$_{\text{ESA+AUG+KNOW}}$" applies knowledge to detect conflicting events. (GA) means that we use gold event arguments instead of approximated ones from SRL.

Grishman, 2010; Hong et al., 2011; Huang and Riloff, 2012a; Huang and Riloff, 2012b). Li et al. (2013) presented a structured perceptron model to detect triggers and arguments jointly. Attempts have also been made to use a Distributional Semantic Model (DSM) to represent events (Goyal et al., 2013). A shortcoming of DSMs is that they ignore the structure within the context, thus reducing the distribution to a bag of words. In our work, we preserve event structure via structured vector representations constructed from event components.

Event co-reference is much less studied in comparison to the large body of work on entity co-reference. Our work follows the event co-reference definition in Hovy et al. (2013). All previous work on event co-reference except Cybulska and Vossen (2012) deals only with full co-reference. Early works (Humphreys et al., 1997; Bagga and Baldwin, 1999) performed event co-reference on scenario specific events. Both Naughton (2009) and Elkhlifi and Faiz (2009) worked on sentence-level co-reference, which is closer to the definition of Danlos and Gaiffe (2003). Pradhan et al. (2007) dealt with both entity and event coreference by taking a three-layer approach. Chen and Ji (2009) proposed a clustering algorithm using a maximum entropy model with a range of features. Bejan and Harabagiu (2010) built a class of nonparametric Bayesian models using a (potentially infinite) number of features to resolve both within and cross document event co-reference. Lee et al. (2012) formed a system with deterministic layers to make co-reference decisions iteratively while jointly resolving entity and event co-reference. More recently, Hovy et al. (2013) presented an unsupervised model to capture semantic relations and co-reference resolution, but they did not show quantitatively how well their system performed in each of these two cases. Huang et al. (2016) also considered

ACE (Cross-Validation)	MUC	B^3	CEAF$_e$	BLANC	AVG
SSED + Supervised$_{\text{Extend}}$	**47.1**	59.9	58.7	44.4	**52.5**
SSED + MSEP-Coref$_{\text{ESA+AUG+KNOW}}$	42.1	**60.3**	**59.0**	44.1	51.4
MSEP-EMD + MSEP-Coref$_{\text{ESA+AUG+KNOW}}$	40.2	58.6	57.4	43.8	50.0
TAC-KBP (Test Data)	MUC	B^3	CEAF$_e$	BLANC	AVG
TAC-TOP	—	—	—	—	**39.1**
SSED + Supervised$_{\text{Extend}}$	**34.9**	44.2	39.6	**37.1**	**39.0**
SSED + MSEP-Coref$_{\text{ESA+AUG+KNOW}}$	33.1	**44.6**	**39.7**	36.8	38.5
MSEP-EMD + MSEP-Coref$_{\text{ESA+AUG+KNOW}}$	30.2	43.9	38.7	35.7	37.1

Table 6: Event Co-reference End-To-End Results.

	Train	Test	MSEP	Supervised
Event Detection				Span+Type F1
In Domain	NW	NW	58.5	**63.7**
Out of Domain	DF	NW	**55.1**	54.8
In Domain	DF	DF	57.9	**62.6**
Out of Domain	NW	DF	**52.8**	52.3
Event Co-reference				AVG F1
In Domain	NW	NW	73.2	**73.6**
Out of Domain	DF	NW	**71.0**	70.1
In Domain	DF	DF	68.6	**68.9**
Out of Domain	NW	DF	**67.9**	67.0

Table 7: **Domain Transfer Results.** We conduct the evaluation on TAC-KBP corpus with the split of newswire (NW) and discussion form (DF) documents. Here, we choose MSEP-EMD and MSEP-Coref$_{\text{ESA+AUG+KNOW}}$ as the MSEP approach for event detection and co-reference respectively. We use SSED and Supervised$_{\text{Base}}$ as the supervised modules for comparison. For event detection, we compare F1 scores of span plus type match while we report the average F1 scores for event co-reference.

the problem of event clustering. They represented event structures based on AMR (Abstract Meaning Representation) and distributional semantics, and further generated event schemas composing event triggers and argument roles. Recently, TAC has organized Event Nugget Detection and Co-reference Evaluations, resulting in interesting works, some of which contributed to our comparisons (Liu et al., 2015; Mitamura et al., 2015; Hsi et al., 2015; Sammons et al., 2015).

6 Conclusion

This paper proposes a novel event detection and co-reference approach with minimal supervision, addressing some of the key issues slowing down progress in research on events, including the difficulty to annotate events and their relations. At the heart of our approach is the design of structured vector representations for events which, as we show, supports a good level of generalization within and across domains. The resulting approach outperforms state-of-art supervised methods on some of the key metrics, and adapts significantly better to a new domain. One of the key research directions is to extend this unsupervised approach to a range of other relations among events, including temporal and causality relations, as is (Do et al., 2011; Do et al., 2012).

Acknowledgments

The authors would like to thank Eric Horn for comments that helped to improve this work. This material is based on research sponsored by the US Defense Advanced Research Projects Agency (DARPA) under agreements FA8750-13-2-000 and HR0011-15-2-0025. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

References

A. Bagga and B. Baldwin. 1998. Algorithms for scoring coreference chains. In *MUC-7*.

A. Bagga and B. Baldwin. 1999. Cross-document event coreference: Annotations, experiments, and observations. In *Proceedings of the Workshop on Coreference and its Applications*.

C. A. Bejan and S. Harabagiu. 2010. Unsupervised event coreference resolution with rich linguistic features. In *ACL*.

E. Bengtson and D. Roth. 2008. Understanding the value of features for coreference resolution. In *EMNLP*.

O. Bronstein, I. Dagan, Q. Li, H. Ji, and A. Frank. 2015. Seed-based event trigger labeling: How far can event descriptions get us? In *ACL*.

P. Brown, V. Della Pietra, P. deSouza, J. Lai, and R. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*.

M. Chang, L. Ratinov, D. Roth, and V. Srikumar. 2008. Importance of semantic represenation: Dataless classification. In *AAAI*.

Z. Chen and H. Ji. 2009. Graph-based event coreference resolution. In *Proceedings of the Workshop on Graph-based Methods for Natural Language Processing*.

Z. Chen, H. Ji, and R. Haralick. 2009. A pairwise event coreference model, feature impact and evaluation for event coreference resolution. In *Proceedings of the Workshop on Events in Emerging Text Types*.

Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *ACL*.

X. Cheng and D. Roth. 2013. Relational inference for wikification. In *EMNLP*.

A. Cybulska and P. Vossen. 2012. Using semantic relations to solve event coreference in text. In *Proceedings of the Workshop on Semantic relations*.

L. Danlos and B. Gaiffe. 2003. Event coreference and discourse relations. *Philosophical Studies Series*.

Q. Do, Y. S. Chan, and D. Roth. 2011. Minimally supervised event causality extraction. In *EMNLP*.

Q. Do, W. Lu, and D. Roth. 2012. Joint inference for event timeline construction. In *EMNLP*.

A. Elkhlifi and R. Faiz. 2009. Automatic annotation approach of events in news articles. *International Journal of Computing & Information Sciences*.

Evgeniy Gabrilovich and Shaul Markovitch. 2009. Wikipedia-based semantic interpretation for natural language processing. *J. Artif. Int. Res.*, 34(1):443–498, March.

K. Goyal, S. K. Jauhar, H. Li, M. Sachan, S. Srivastava, and E. Hovy. 2013. A structured distributional semantic model for event co-reference. In *ACL*.

Y. Hong, J. Zhang, B. Ma, J. Yao, G. Zhou, and Q. Zhu. 2011. Using cross-entity inference to improve event extraction. In *ACL*.

E. Hovy, T.o Mitamura, F. Verdejo, J. Araki, and A. Philpot. 2013. Events are not simple: Identity, non-identity, and quasi-identity. In *NAACL-HLT*.

A. Hsi, J. Carbonell, and Y. Yang. 2015. Modeling event extraction via multilingual data sources. In *TAC*.

R. Huang and E. Riloff. 2012a. Bootstrapped training of event extraction classifiers. In *EACL*.

R. Huang and E. Riloff. 2012b. Modeling textual cohesion for event extraction. In *AAAI*.

L. Huang, T. Cassidy, X. Feng, H. Ji, C. R. Voss, J. Han, and A. Sil. 2016. Liberal event extraction and event schema induction. In *ACL*.

K. Humphreys, R. Gaizauskas, and S. Azzam. 1997. Event coreference for information extraction. In *Proceedings of Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*.

H. Ji and R. Grishman. 2008. Refining event extraction through cross-document inference. In *ACL*.

H. Lee, M. Recasens, A. Chang, M. Surdeanu, and D. Jurafsky. 2012. Joint entity and event coreference resolution across documents. In *EMNLP*.

O. Levy and Y. Goldberg. 2014. Dependencybased word embeddings. In *ACL*.

Q. Li, H. Ji, and L. Huang. 2013. Joint event extraction via structured prediction with global features. In *ACL*.

S. Liao and R. Grishman. 2010. Using document level cross-event inference to improve event extraction. In *ACL*.

Z. Liu, T. Mitamura, and E. Hovy. 2015. Evaluation algorithms for event nugget detection: A pilot study. In *Proceedings of the Workshop on Events at the NAACL-HLT*.

X. Luo. 2005. On coreference resolution performance metrics. In *EMNLP*.

T. Mikolov, W. Yih, and G. Zweig. 2013. Linguistic regularities in continuous space word representations. In *NAACL*.

T. Mitamura, Y. Yamakawa, S. Holm, Z. Song, A. Bies, S. Kulick, and S. Strassel. 2015. Event nugget annotation: Processes and issues. In *Proceedings of the Workshop on Events at NAACL-HLT*.

M. Naughton. 2009. *Sentence Level Event Detection and Coreference Resolution*. Ph.D. thesis, National University of Ireland, Dublin.

V. Ng and C. Cardie. 2002. Improving machine learning approaches to coreference resolution. In *ACL*.

NIST. 2005. The ACE evaluation plan.

S. Pradhan, L. Ramshaw, R. Weischedel, J. MacBride, and L. Micciulla. 2007. Unrestricted coreference: Identifying entities and events in ontonotes. In *ICSC*.

V. Punyakanok, D. Roth, W. Yih, and D. Zimak. 2004. Semantic role labeling via integer linear programming inference. In *COLING*.

M. Recasens and E. Hovy. 2011. Blanc: Implementing the rand index for coreference evaluation. *Natural Language Engineering*, 17(04):485–510.

M. Sammons, H. Peng, Y. Song, S. Upadhyay, C.-T. Tsai, P. Reddy, S. Roy, and D. Roth. 2015. Illinois ccg tac 2015 event nugget, entity discovery and linking, and slot filler validation systems. In *TAC*.

Y. Song and D. Roth. 2014. On dataless hierarchical text classification. In *AAAI*.

V. Stoyanov, C. Cardie, N. Gilbert, E. Riloff, D. Buttler, and D. Hysom. 2010. Coreference resolution with reconcile. In *ACL*.

M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th conference on Message understanding*.

R. Zhao, Q. Do, and D. Roth. 2012. A robust shallow temporal reasoning system. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT Demo)*.