Joint Extraction of Events and Entities within a Document Context

Bishan Yang
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA, 15213
bishan@cs.cmu.edu

Tom Mitchell
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA, 15213
tom.mitchell@cs.cmu.edu

Abstract

Events and entities are closely related; entities are often actors or participants in events and events without entities are uncommon. The interpretation of events and entities is highly contextually dependent. Existing work in information extraction typically models events separately from entities, and performs inference at the sentence level, ignoring the rest of the document. In this paper, we propose a novel approach that models the dependencies among variables of events, entities, and their relations, and performs joint inference of these variables across a document. The goal is to enable access to document-level contextual information and facilitate context-aware predictions. We demonstrate that our approach substantially outperforms the state-of-the-art methods for event extraction as well as a strong baseline for entity extraction.

1 Introduction

Events are things that happen or occur; they involve entities (people, objects, etc.) who perform or are affected by the events and spatio-temporal aspects of the world. Understanding events and their descriptions in text is necessary for any generally-applicable machine reading systems. It is also essential in facilitating practical applications such as news summarization, information retrieval, and knowledge base construction.

The interpretation of event descriptions is highly contextually dependent. To make correct predictions, a model needs to account for mentions of events and entities together with the discourse context. Consider, for example, the following excerpt from a news report:

> "On *Thursday*, there was a massive *U.S.* **aerial bombardment** in which more than 300 *Tomahawk cruise missiles* rained down on *Baghdad*. *Earlier Saturday*, *Baghdad* was again **targeted**. ..."

The excerpt describes two U.S. attacks on Baghdad. The two event anchors (triggers) are boldfaced and the mentions of entities and spatio-temporal information are italicized. The first event anchor "aerial bombardment" along with its surrounding entity mentions — "U.S.", "Tomahawk cruise missiles", and "Baghdad", describe an attack from the U.S. on Baghdad with Tomahawk cruise missiles being the weapon. The second sentence on its own contains little event-related information, but together with the context of the previous sentence, it indicates another U.S. attack on Baghdad.

State-of-the-art event extraction systems have difficulties inferring such information due to two main reasons. First, they extract events and entities in separate stages: entities such as people, organization, and locations are first extracted by a named entity tagger, and then these extracted entities are used as inputs for extracting events and their arguments (Li et al., 2013). This often causes errors to propagate. In the above example, if the entity tagger mistakenly identifies "Baghdad" as a person, then the event extractor will fail to extract "Baghdad" as the place where the attack happened. In fact, previous work (Li et al., 2013) observes that using previously extracted entities in event extraction results in

Proceedings of NAACL-HLT 2016, pages 289–299,
San Diego, California, June 12-17, 2016.

a substantial decrease in performance compared to using gold-standard entity information.

Second, most existing work extracts events independently from each individual sentence, ignoring the rest of the document (Li et al., 2013; Judea and Strube, 2015; Nguyen and Grishman, 2015). Very few attempts have been made to incorporate document context for event extraction. Ji and Grishman (2008) model the information flow in two stages: the first stage trains classifiers for event triggers and arguments within each sentence; the second stage applies heuristic rules to adjust the classifiers' outputs to satisfy document-wide (or document-cluster-wide) consistency. Liao and Grishman (2010) further improved the rule-based inference by training additional classifiers for event triggers and arguments using document-level information. Both approaches only propagate the highly confident predictions from the first stage to the second stage. To the best of our knowledge, there is no unified model that jointly extracts events from sentences across the whole document.

In this paper, we propose a novel approach that simultaneously extracts events and entities within a document context.[1] We first decompose the learning problem into three tractable subproblems: (1) learning the dependencies between a single event and all of its potential arguments, (2) learning the co-occurrence relations between events across the document, and (3) learning for entity extraction. Then we combine the learned models for these subproblems into a joint optimization framework that simultaneously extracts events, semantic roles, and entities in a document. In summary, our main contributions are:

1. We propose a structured model for learning within-event structures that can effectively capture the dependencies between an event and its arguments, and between the semantic roles and entity types for the arguments.

2. We introduce a joint inference framework that combines probabilistic models of within-event structures, event-event relations, and entity extraction for joint extraction of the set of entities and events over the whole document.

3. We conduct extensive experiments on the Automatic Content Extraction (ACE) corpus, and show that our approach significantly outperforms the state-of-the-art methods for event extraction and a strong baseline for entity extraction.

2 Task Definition

We adopt the ACE definition for entities ((LDC), 2005a) and events ((LDC), 2005b):

- **Entity mention**: An entity is an object or set of objects in the world. An entity mention is a reference to an entity in the form of a noun phrase or a pronoun.

- **Event trigger**: the word or phrase that clearly expresses its occurrence. Event triggers can be verbs, nouns, and occasionally adjectives like "dead" or "bankrupt".

- **Event argument**: event arguments are entities that fill specific roles in the event. They mainly include participants (i.e., the entities that are involved in the event) and general event attributes such as place and time, and some event-type-specific attributes that have certain values (e.g., JOB-TITLE, CRIME).

We are interested in extracting entity mentions, event triggers, and event arguments. We consider ACE entity types PER, ORG, GPE, LOC, FAC, VEH, WEA and ACE VALUE and TIME expressions[2], and focus on 33 ACE event subtypes, each of which has its own set of semantic roles for the potential arguments. There are 35 such roles in total, but we collapse 8 of them that are time-related (e.g., TIME-HOLDS, TIME-AT-END) into one, because most of these roles have very few training examples. Figure 2 shows an example of ACE annotations for events and entities in a sentence. Note that not every entity mention in the sentence is involved in events and a single entity mention can be associated with multiple events.

[1] The code for our system is available at https://github.com/bishanyang/EventEntityExtractor.

[2] To simplify notation, we include values and times when referring to entities in the rest of the paper.

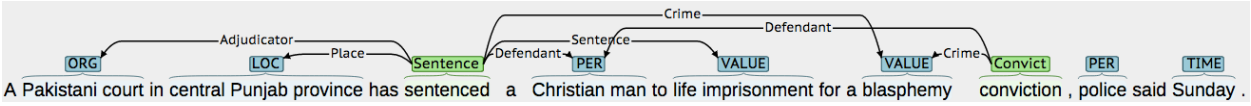

Figure 1: An example of ACE annotations of events and entities. The event triggers and the entity mentions are marked in different colors. Each event trigger has an event subtype marked above it and each entity mention has an entity type marked above it. Each event trigger evokes an event with semantic roles that are filled by entity mentions. The roles are marked on the links between event trigger and entity mentions. For example, "conviction" evokes a CONVICT event, and has the CRIME and DEFENDANT roles filled by "blasphemy" and "Christian man" respectively.

3 Approach

In this section, we describe our approach for joint extraction of events and entities within a document context. We first decompose the learning problem into three tractable subproblems: learning within-event structures, learning event-event relations, and learning for entity extraction. We will describe the probabilistic models for learning these subproblems. Then we present a joint inference framework that integrates these learned models into a single model to jointly extract events and entities across a document.

3.1 Learning Within-event Structures

As described in Section 2, a mention of an event consists of an event trigger and a set of event arguments. Each event argument is also an entity mention with an entity type. In the following, we develop a probabilistic model to learn such dependency structure for each individual event mention.

Given a document x, we first generate a set of event trigger candidates $\mathcal{T}$ and a set of entity candidates $\mathcal{N}$.[3] For each trigger candidate $i \in \mathcal{T}$, we associate it with a discrete variable t_i that takes values from the 33 ACE event types and a NONE class indicating other events or no events. Denote the set of entity candidates that are potential arguments for trigger candidate i as $\mathcal{N}_i$.[4] For each $j \in \mathcal{N}_i$, we associate it with a discrete variable r_{ij} which models the event-argument relation between trigger candidate i and entity candidate j. It takes values from 28 semantic roles and a NONE class indicating invalid roles. Each argument candidate j is also associated with an entity type variable a_j, which takes values from 9 entity types and a NONE class indicating invalid entity types.

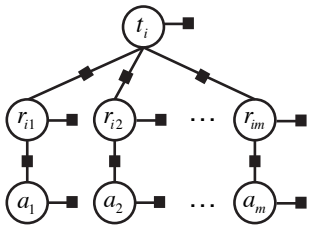

Figure 2: A factor graph representation of the within-event model, relating the event type t_i of trigger candidate i to the role type r_{ij} of each argument candidate j along with its entity type a_j.

We define the joint distribution of variables t_i, $\mathbf{r}_{i\cdot} = \{r_{ij}\}_{j\in\mathcal{N}_i}$, and $\mathbf{a}_{\cdot} = \{a_j\}_{j\in\mathcal{N}_i}$ conditioned on the observations, which can be factorized according to the factor graph shown in Figure 2:

$$p_{\boldsymbol{\theta}}(t_i, \mathbf{r}_{i\cdot}, \mathbf{a}_{\cdot}|i, \mathcal{N}_i, x) \propto \exp\Bigg(\boldsymbol{\theta}_1^T f_1(t_i, i, x) + \sum_{j\in\mathcal{N}_i} \boldsymbol{\theta}_2^T f_2(r_{ij}, i, j, x) + \sum_{j\in\mathcal{N}_i} \boldsymbol{\theta}_3^T f_3(t_i, r_{ij}, i, j, x) + \sum_{j\in\mathcal{N}_i} \boldsymbol{\theta}_4^T f_4(a_j, j, x) + \sum_{j\in\mathcal{N}_i} \boldsymbol{\theta}_5^T f_5(r_{ij}, a_j, j, x)\Bigg) \quad (1)$$

where $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_5$ are vectors of parameters that need to be estimated, and $f_1, ..., f_5$ are different forms of feature functions which we will describe later.

Note that not all configurations of the variables are valid in our model. Based on the definitions in Section 2, each event type takes arguments with certain semantic roles. For example, the arguments of the event MARRY can only play the roles of

[3] We describe how to extract these candidates in Section 4.

[4] In this paper we only consider entity mentions that are in the same sentence as the trigger to be potential event arguments due to the ACE annotations. However, our model is general and can handle event-argument relations across sentences with appropriate features.

PERSON, TIME, and PLACE. In addition, a NONE event type should not take any arguments. Similarly, each semantic role should be filled with entities with compatible types. For example, the PERSON role type can only be filled with an entity of type PER. However, a NONE role type can be filled with an entity of any type. To account for these compatibility constraints, we enforce the probabilities of all invalid configurations to be zero.

Features. f_1, f_2, and f_4 are unary feature functions that depend on trigger variable t_i, argument variable r_{ij}, and entity variable a_j respectively. We construct a set of features for each feature function (see Table 1). Many of these features overlap with those used in previous work (Li et al., 2013; Li et al., 2014), except for the word embedding features for triggers and the features for entities which are derived from multiple entity resources. f_3 and f_5 are pairwise feature functions that depend on trigger-argument pair (t_i, r_{ij}) and argument-entity pair (r_{ij}, a_j) respectively. We consider simple indicator functions $\mathbb{1}_{t,r}$ and $\mathbb{1}_{r,a}$ as features ($\mathbb{1}_y(x)$ equals 1 when $x = y$ and 0 otherwise).

Training. For model training, we find the optimal parameters $\boldsymbol{\theta}$ using the maximum-likelihood estimates with an L2 regularization:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) - \lambda||\boldsymbol{\theta}||_2^2$$

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_i \log p(t_i, \mathbf{r}_{i\cdot}, \mathbf{a}_\cdot | i, \mathcal{N}_i, x)$$

We use L-BFGS to optimize the training objective. To calculate the gradient, we use the sum-product algorithm to compute the exact marginals for the unary cliques t_i, r_{ij}, a_j and the pairwise cliques $(t_i, r_{ij}), (r_{ij}, a_j)$. Typically the training complexity for graphical models with unary and pairwise cliques is quadratic in the size of the label set. However, the complexity of our model is much lower than that since we only need to compute the joint distributions over valid variable configurations. Denote the number of event subtypes as T, the number of event argument roles as N, the average number of argument roles for each event subtype as k_1, the average number of entity types for each event argument as k_2, and the average number of argument candidates for each trigger candidate as M. The complexity of computing the joint distribution is $O(M \times (k_1 T + k_2 N))$, and k_1 and k_2 are expected to be small in practice ($k_1 = 6, k_2 = 3$ in ACE).

3.2 Learning Event-Event Relations

So far we have described a model for learning structures for a single event. However, the inference of the event types for individual events may depend on other events that are mentioned in the document. For example, an ATTACK event is more likely to occur with INJURE and DIE events than with life events like MARRY and BORN. In order to capture this intuition, we develop a pairwise model of event-event relations in a document.

Our training data consists of all pairs of trigger candidates that co-occur in the same sentence or are connected by a coreferent subject/object if they are in different sentences.[5] We want to propagate information between these trigger pairs since they are more likely to be related.

Formally, given a trigger candidate pair (i, i'), we estimate the probabilities for their event types $(t_i, t_{i'})$ as

$$p_{\boldsymbol{\phi}}(t_i, t_{i'} | x, i, i') \propto \exp\left(\boldsymbol{\phi}^T g(t_i, t_{i'}, x, i, i')\right) \quad (2)$$

where $\boldsymbol{\phi}$ is a vector of parameters and g is a feature function that depends on the trigger candidate pair and their context. We consider both trigger-specific features and relational features. For trigger-specific features, we use the same trigger features listed in Table 1. For relational features, we consider for each pair of trigger candidates: (1) whether they are connected by a conjunction dependency relation (based on dependency parsing); (2) whether they share a subject or an object (based on dependency parsing and coreference resolution); (3) whether they have the same head word lemma; (4) whether they share a semantic frame based on FrameNet. During training, we use L-BFGS to compute the maximum-likelihood estimates of ϕ.

3.3 Entity Extraction

For entity extraction, we trained a standard linear-chain Conditional Random Field (CRF) (Lafferty et al., 2001) using the BIO scheme (i.e., identifying the **B**eginning, the **I**nside and the **O**utside of the

[5] We use the Stanford coreference system (Lee et al., 2013) for within-document entity coreference.

Category	Type	Features
Trigger	*Lexical resources:* WordNet Nomlex FrameNet Word2Vec	1. lemmas of the words in the trigger mention 2. nominalization of the words based on Nomlex (Macleod et al., 1998) 3. context words within a window of size 2 4. similarity features between the head word and a list of trigger seeds based on WordNet (Bronstein et al., 2015) 5. semantic frames that associate with the head word and its p-o-s tag based on FrameNet (Li et al., 2014) 6. pre-trained vector for the head word (Mikolov et al., 2013)
	Syntactic resources: Stanford parser	7. dependency edges involving the head word, both lexicalized and unlexicalized 8. whether the head word is a pronoun
Argument	*Lexical resources:* WordNet	1. lemmas of the words in the entity mention 2. lemmas of the words in the trigger mention 3. words between the entity mention and the trigger mention
	Syntactic resources: Stanford parser	4. the relative position of the entity mention to the trigger mention (before, after, or contain) 5. whether the entity mention and the trigger mention are in the same clause 6. the shortest dependency paths between the entity mention and the trigger mention
Entity	*Entity resources:* Stanford NER NELL KB	1. Gender and animacy attributes of the entity mention 2. Stanford NER type for the entity mention 3. Semantic type for the entity mention based on the NELL knowledge base (Mitchell et al., 2015) 4. Predicted entity type and confidence score for the entity mention output by the entity extractor described in Section 3.3

Table 1: Features for event triggers, event arguments, and entity mentions.

text segments). We use features that are similar to those from previous work (Ratinov and Roth, 2009): (1) current words and part-of-speech tags; (2) context words in a window of size 2; (3) word type such as all-capitalized, is-capitalized, and all-digits; (4) Gazetteer-based entity type if the current word matches an entry in the gazetteers collected from Wikipedia (Ratinov and Roth, 2009). In addition, we consider pre-trained word embeddings (Mikolov et al., 2013) as dense features for each word in order to improve the generalizability of the model.

3.4 Joint Inference

Our end goal is to extract coherent event mentions and entity mentions across a document. To achieve this, we propose a joint inference approach that allows information flow among the three local models and finds globally-optimal assignments of all variables, including the trigger variables $\mathbf{t}$, the argument role variables $\mathbf{r}$, and the entity variables $\mathbf{a}$. Specifically, we define the following objective:

$$\max_{\mathbf{t},\mathbf{r},\mathbf{a}} \sum_{i \in \mathcal{T}} E(t_i, \mathbf{r}_{i\cdot}, \mathbf{a}_\cdot) + \sum_{i,i' \in \mathcal{T}} R(t_i, t_{i'}) + \sum_{j \in \mathcal{N}} D(a_j) \quad (3)$$

The first term is the sum of confidence scores for individual event mentions based on the parameter estimates from the within-event model. $E(t_i, \mathbf{r}_{i\cdot}, \mathbf{a}_\cdot)$ can be further decomposed into three parts.

$$E(t_i, \mathbf{r}_{i\cdot}, \mathbf{a}_\cdot) = \log p_{\boldsymbol{\theta}}(t_i | i, \mathcal{N}_i, x) + \sum_{j \in \mathcal{N}_i} \log p_{\boldsymbol{\theta}}(t_i, r_{ij} | i, \mathcal{N}_i, x) + \sum_{j \in \mathcal{N}_i} \log p_{\boldsymbol{\theta}}(r_{ij}, a_j | i, \mathcal{N}_i, x)$$

The second term is the sum of confidence scores for event relations based on the parameter estimates from the pairwise event model, where $R(t_i, t_{i'}) = \log p_{\boldsymbol{\phi}}(t_i, t_{i'} | i, i', x)$. The third term is the sum of confidence scores for entity mentions, where $D(a_j) = \log p_\psi(a_j | j, x)$ and $p_\psi(a_j | j, x)$ is the marginal probability derived from the linear-chain CRF described in Section 3.3. The optimization is subjected to agreement constraints that enforce the overlapping variables among the three components to agree on their values.

The joint inference problem can be formulated as an integer linear program (ILP). To solve it efficiently, we find solutions for the relaxation of

the problem using a dual decomposition algorithm AD^3 (Martins et al., 2011). AD^3 has been shown to be orders of magnitude faster than a general purpose ILP solver in practice (Das et al., 2012). It is also particularly suitable for our problem since it involves decompositions that have many overlapping simple factors. We observed that AD^3 recovers the exact solutions for all the test documents in our experiments and the runtime for labeling each document is only three seconds in average in a 64-bit machine with two 2GHz CPUs and 8GB of RAM.

4 Experiments

We conduct experiments on the ACE2005 corpus.[6] It contains text documents from a variety of sources such as newswire reports, weblogs, and discussion forums. We use the same data split as in Li et al. (2013). Table 2 shows the data statistics.

We adopt the evaluation metrics for events as defined in Li et al. (2013). An event trigger is correctly identified if its offsets match those of a gold-standard trigger; and it is correctly classified if its event subtype (33 in total) also match the subtype of the gold-standard trigger. An event argument is correctly identified if its offsets and event subtype match those of any of the reference argument mentions in the document; and it is correctly classified if its semantic role (28 in total) is also correct. For entities, a predicted mention is correctly extracted if its head offsets and entity type (9 in total) match those of the reference entity mention.

Note that our approach requires entity mention candidates and event trigger candidates as input. Instead of enumerating all possible text spans, we generate high-quality entity mentions from the k-best predictions of our CRF entity extractor (in Section 3.3).[7] Similarly, we train a CRF for event trigger extraction using the same features except for the gazetteers, and generate trigger candidates based on the k-best predictions. We set $k = 50$ for entities and $k = 10$ for event triggers based on performance on the development set. They cover 92.3% of the gold-standard entity mentions and 96.3% of the gold-standard event triggers in the test set.

[6] http://www.itl.nist.gov/iad/mig/tests/ace/2005/

[7] During training, we randomly split the training data into 10 parts and consider the k-best predictions for each part.

	Train	Dev	Test
Documents	529	40	30
Sentences	14,837	863	672
Triggers	4,337	497	438
Arguments	7,768	933	911
Entity Mentions	48,797	3,917	4,184

Table 2: Statistics of the ACE2005 dataset.

4.1 Results

Event Extraction. We compare the proposed models WITHINEVENT (in Section 3.1) and JOINTEVENTENTITY (in Section 3.4) with two strong baselines. One is JOINTBEAM (Li et al., 2013), a state-of-the-art event extractor that uses a structured perceptron with beam search for sentence-level joint extraction of event triggers and arguments. The other is STAGEDMAXENT, a typical two-stage approach that detects event triggers first and then event arguments. We use the same event trigger candidates and entity mention candidates as input to all the comparing models except for JOINTBEAM, because JOINTBEAM only extracts event mentions and assumes entity mentions are given. We consider a realistic experimental setting where no gold-standard annotations are available for entities during testing. To obtain results from JOINTBEAM, we ran the actual system[8] used in Li et al. (2013) using the entity mentions output by our CRF-based entity extractor.

Table 3 shows the average[9] precision, recall, and F1 score for event triggers and event arguments. We can see that our WITHINEVENT model, which explicitly models the trigger-argument dependencies and argument-role-entity-type dependencies, outperforms the MaxEnt pipeline, especially in event argument extraction. This shows that modeling the trigger-argument dependencies is effective in reducing error propagation.

Comparing to the state-of-the-art event extractor JOINTBEAM, the improvements introduced by WITHINEVENT are substantial in both event triggers and event arguments. We believe there are two main reasons: (1) WITHINEVENT considers all possible joint trigger/argument label assignments, whereas

[8] https://github.com/oferbr/BIU-RPI-Event-Extraction-Project

[9] We report the micro-average scores as in previous work (Li et al., 2013).

Model	Event Trigger Identification			Event Trigger Classification			Event Argument Identification			Argument Role Classification		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
JOINTBEAM (Li et al., 2013)	76.6	58.7	66.5	74.0	56.7	64.2	74.6	25.5	38.0	68.8	23.5	35.0
STAGEDMAXENT	73.9	**66.5**	70.0	70.4	63.3	66.7	**75.7**	20.2	31.9	**71.2**	19.0	30.0
WITHINEVENT	76.9	63.8	69.7	74.7	62.0	67.7	72.4	37.2	49.2	69.9	35.9	47.4
JOINTEVENTENTITY	**77.6**	65.4	**71.0***	**75.1**	63.3	**68.7**	73.7	**38.5**	**50.6***	70.6	**36.9**	**48.4***

Table 3: Event extraction results on the ACE2005 test set. $*$ indicates that the difference in F1 compared to the second best model (WITHINEVENT) is statistically significant ($p < 0.05$).

Model	Trigger	Arg
CROSS-DOC (Ji and Grishman, 2008)	67.3	42.6
CNN (Nguyen and Grishman, 2015)	67.6	-
JOINTEVENTENTITY	**68.7**	**48.4**

Table 4: Comparison of the results (F1 score) of JOINTEVENTENTITY and the best known results on ACE event trigger classification and argument role classification.

Model	P	R	F1
CRFENTITY	**85.5**	73.5	79.1
JOINTEVENTENTITY	82.4	**79.2**	**80.7***

Table 5: Entity extraction results on the ACE2005 test set. $*$ indicates statistical significance ($p < 0.05$).

JOINTBEAM considers only a subset of the possible assignments based on a heuristic beam search. More specifically, when predicting labels for token i, JointBeam considers only the K-best ($K = 4$ in their paper) partial trigger/argument label configurations for the previous $i - 1$ tokens. As the length of the sentence increases, a large amount of information will be thrown away. (2) WITHINEVENT models argument-role-entity-type dependencies, whereas JOINTBEAM assumes the entity types are given. This can cause error propagation.

JOINTEVENTENTITY provides the best performance among all the models on all evaluation categories. It boosts both precision and recall compared to WITHINEVENT.[10] This demonstrates the advantages of JOINTEVENTENTITY in allowing information propagation across event mentions and entity mentions and making more context-aware and semantically coherent predictions.

We also compare the results of JOINTEVENTENTITY with the best known results on the ACE event extraction task in Table 4. CROSS-DOC (Ji and Grishman, 2008) performs cross-document inference of events using document clustering information, and CNN (Nguyen and Grishman, 2015) is a convolutional neural network for extracting event triggers at the sentence level. We see that JOINTEVENTENTITY outperforms both models and achieves new state-of-the-art results for event trigger and argument extraction in an end-to-end evaluation setting.

[10] All significance tests reported in this paper were computed using the paired bootstrap procedure (Berg-Kirkpatrick et al., 2012) with 10,000 samples of the test documents.

Model	PER	GPE	ORG	TIME
CRFENTITY	85.1	87.0	65.4	78.4
JOINTEVENTENTITY	**87.1**	87.0	**70.2**	**80.2**

Table 6: Entity extraction results (F1 score) per entity type.

Entity Extraction. In addition to extracting event mentions, JOINTEVENTENTITY also extracts entity mentions. We compare its output with the output of a strong entity extraction baseline CRFENTITY (described in Section 3.3). Table 5 shows the (micro-)average precision, recall, and F1 score. We see that JOINTEVENTENTITY introduces a significant improvement in recall and F1. Table 6 further shows the F1 score for four major entity types PER, GPE, ORG, and TIME in ACE. The promising improvements indicate that joint modeling of events and entities allows for more accurate predictions about not only events but also entities.

4.2 Error Analysis

Table 7 divides the errors made by JOINTEVENTENTITY based on different subtasks and the classification error types in each task. For event triggers, the majority of the errors relates to missing triggers and only 3.7% involves misclassified event types (e.g., a DEMONSTRATION event is mistaken for a TRANSPORT event). Among the missing triggers, we examine the cases where the event types are correctly identified in a sentence but with in-

Error Type	Missing	Spurious	Misclassified
TRIGGER	62.1%	34.2%	3.7%
ARGUMENT	71.2%	24.7%	4.1%
ENTITY	43.4%	30.5%	26.1%

Table 7: Classification of errors made by JOINTEVENTENTITY.

correct triggers and find that there are only 5% of such cases. For event arguments, the majority of the errors relates to missing arguments and only 4.1% is about misclassified argument roles. Among the missing event arguments, 10% of them has correctly identified entity types.

In general, the errors for event extraction are commonly due to three reasons: (1) Lexical sparsity. For example, in the sentence "At least three members of a family ... were **hacked** to death ...", our model fails to detect that "hacked" triggers an ATTACK event, because it has never seen "hacked" with this sense during training. Using WordNet and pretrained word vectors may alleviate the sparsity issue. It is also important to disambiguate word senses in context. (2) Shallow understanding of context, especially long-range context. For example, given the sentence "**She** is being held on 50,000 dollars bail on a charge of first-degree reckless **homicide** ...", the model detects that "homicide" triggers an event, but fails to detect that "She" refers to the AGENT who committed the homicide. This is mainly due to the complex long-distance dependency between the trigger and the argument. (3) Use of complex language such as metaphor, idioms, and sarcasm. Addressing these phenomena is in general difficult since it requires richer background knowledge and more sophisticated inference.

For entity extraction, we find that integrating event information into entity extraction successfully improves recall and F1. However, since the ACE dataset is restricted to a limited set of events, a large portion of the sentences does not contain any event triggers and event arguments that are of interest. For these sentences, there is little or no benefit of joint modeling. We also find that some entity misclassification errors can be avoided if entity coreference information is available. We plan to investigate coreference resolution as an additional component to our joint model in future work.

5 Related Work

Event extraction has been mainly studied using the ACE data (Doddington et al., 2004) and biomedical data for the BioNLP shared tasks (Kim et al., 2009). To reduce task complexity, early work employs a pipeline of classifiers that extracts event triggers first, and then determines their arguments (Ahn, 2006; Björne et al., 2009). Recently, Convolutional Neural Networks have been used to improve the pipeline classifiers (Nguyen and Grishman, 2015; Chen et al., 2015). As pipeline approaches suffer from error propagation, researchers have proposed methods for joint extraction of event triggers and arguments, using either structured perceptron (Li et al., 2013), Markov Logic (Poon and Vanderwende, 2010), or dependency parsing algorithms (McClosky et al., 2011). However, existing joint models largely rely on heuristic search to aggressively shrink the search space. One exception is work in Riedel and McCallum (2011), which uses dual decomposition to solve joint inference with runtime guarantees. Our work is similar to Riedel and McCallum (2011). However, there are two main differences: first, our model extracts both event mentions and entity mentions; second, it performs joint inference across sentence boundaries. Although our approach is evaluated on ACE, it can be easily adapted to BioNLP data by using appropriate features for events triggers, argument roles, and entities. We consider this as future work.

There has been work on improving event extraction by exploiting document-level context. Berant et al. (2014) exploits event-event relations, e.g., causality, inhibition, which frequently occur in biological texts. For general texts most work focuses on exploiting temporal event relations (Chambers and Jurafsky, 2008; Do et al., 2012; McClosky and Manning, 2012). For the ACE domain, there is work on utilizing event type co-occurrence patterns to propagation event classification decisions (Ji and Grishman, 2008; Liao and Grishman, 2010). Our model is similar to their work. It models the co-occurrence relations between event types (e.g., a DIE event tends to co-occur with ATTACK events and TRANSPORT events). It can be extended to handle other types of event relations (e.g., causal and temporal) by designing appropriate features. Chambers and

Jurafsky (2009; 2011) learn narrative schemas by linking event verbs that have coreferring syntactic arguments. Our model also adopts this intuition to relate event triggers across sentences. In addition, each event argument is grounded by its entity type (e.g., an entity mention of type PER can only fill roles that can be played by a person).

6 Conclusion

In this paper, we introduce a new approach for automatic extraction of events and entities across a document. We first decompose the learning problem into three tractable subproblems: learning within-event structures, learning event-event relations, and learning for entity extraction. We then integrate these learned models into a single model that performs joint inference of all event triggers, semantic roles for events, and entities across the whole document. Experimental results demonstrate that our approach outperforms the state-of-the-art event extractors by a large margin and substantially improves a strong entity extraction baseline. For future work, we plan to integrate entity and event coreference as additional components into the joint inference framework. We are also interested in investigating the integration of more sophisticated event-event relation models of causality and temporal ordering.

Acknowledgments

This work was supported in part by NSF grant IIS-1250956, and in part by the DARPA DEFT program under contract FA87501320005. We would like to thank members of the CMU NELL group for helpful comments. We also thank the anonymous reviewers for insightful suggestions.

References

David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8. Association for Computational Linguistics.

Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Brad Huang, Christopher D Manning, Abby Vander Linden, Brittany Harding, and Peter Clark. 2014. Modeling biological processes for reading comprehension. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1499–1510. Association for Computational Linguistics.

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in nlp. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 995–1005. Association for Computational Linguistics.

Jari Björne, Juho Heimonen, Filip Ginter, Antti Airola, Tapio Pahikkala, and Tapio Salakoski. 2009. Extracting complex biological events with rich graph-based feature sets. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, pages 10–18. Association for Computational Linguistics.

Ofer Bronstein, Ido Dagan, Qi Li, Heng Ji, and Anette Frank. 2015. Seed-based event trigger labeling: How far can event descriptions get us? In *ACL Volume 2: Short Papers*, pages 372–376. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. 2008. Jointly combining implicit constraints improves temporal ordering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 698–706. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 602–610. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. 2011. Template-based information extraction without the templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 976–986. Association for Computational Linguistics.

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

Dipanjan Das, André FT Martins, and Noah A Smith. 2012. An exact dual decomposition algorithm for shallow semantic parsing with constraints. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume*

2: Proceedings of the Sixth International Workshop on Semantic Evaluation, pages 209–217. Association for Computational Linguistics.

Quang Xuan Do, Wei Lu, and Dan Roth. 2012. Joint inference for event timeline construction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 677–687. Association for Computational Linguistics.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC-2004)*. European Language Resources Association (ELRA).

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: HLT*, pages 254–262. Association for Computational Linguistics.

Alex Judea and Michael Strube. 2015. Event extraction as frame-semantic parsing. *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics (*SEM 2015)*, pages 159–164.

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 the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, pages 1–9. Association for Computational Linguistics.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning (ICML)*, pages 282–289.

Linguistic Data Consortium (LDC). 2005a. English annotation guidelines for entities. `https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-entities-guidelines-v5.6.6.pdf`.

Linguistic Data Consortium (LDC). 2005b. English annotation guidelines for events. `https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf`.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916.

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

Qi Li, Heng Ji, Yu Hong, and Sujian Li. 2014. Constructing information networks using one single model. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 1846–1851. Association for Computational Linguistics.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 789–797. Association for Computational Linguistics.

Catherine Macleod, Ralph Grishman, Adam Meyers, Leslie Barrett, and Ruth Reeves. 1998. Nomlex: A lexicon of nominalizations. In *Proceedings of EURALEX*, volume 98, pages 187–193. Citeseer.

André FT Martins, Mario AT Figeuiredo, Pedro MQ Aguiar, Noah A Smith, and Eric P Xing. 2011. An augmented lagrangian approach to constrained map inference. In *Proceedings of the International Conference on Machine Learning (ICML)*.

David McClosky and Christopher D Manning. 2012. Learning constraints for consistent timeline extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 873–882. Association for Computational Linguistics.

David McClosky, Mihai Surdeanu, and Christopher D Manning. 2011. Event extraction as dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL)*, pages 1626–1635. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems (NIPS)*, pages 3111–3119.

T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2015. Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional

neural networks. In *Proceedings of ACL-IJCNLP 2015 Volume 2: Short Papers*, pages 365–371. Association for Computational Linguistics.

Hoifung Poon and Lucy Vanderwende. 2010. Joint inference for knowledge extraction from biomedical literature. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 813–821. Association for Computational Linguistics.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics.

Sebastian Riedel and Andrew McCallum. 2011. Fast and robust joint models for biomedical event extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–12. Association for Computational Linguistics.