Identifying civilians killed by police with distantly supervised entity-event extraction

Katherine A. Keith, Abram Handler, Michael Pinkham,
Cara Magliozzi, Joshua McDuffie, and **Brendan O'Connor**
College of Information and Computer Sciences
University of Massachusetts Amherst
`kkeith@cs.umass.edu, brenocon@cs.umass.edu`
http://slanglab.cs.umass.edu/

Abstract

We propose a new, socially-impactful task for natural language processing: from a news corpus, extract names of persons who have been killed by police. We present a newly collected police fatality corpus, which we release publicly, and present a model to solve this problem that uses EM-based distant supervision with logistic regression and convolutional neural network classifiers. Our model outperforms two off-the-shelf event extractor systems, and it can suggest candidate victim names in some cases faster than one of the major manually-collected police fatality databases.

Appendix, software, and data are available online at: http://slanglab.cs.umass.edu/PoliceKillingsExtraction/

1 Introduction

The United States government does not keep systematic records of when police kill civilians, despite a clear need for this information to serve the public interest and support social scientific analysis. Federal records rely on incomplete cooperation from local police departments, and human rights statisticians assess that they fail to document thousands of fatalities (Lum and Ball, 2015).

News articles have emerged as a valuable alternative data source. Organizations including The Guardian, The Washington Post, Mapping Police Violence, and Fatal Encounters have started to build such databases of U.S. police killings by manually reading millions of news articles[1]

---

[1] Fatal Encounters director D. Brian Burghart estimates he and colleagues have read 2 million news headlines and ledes to assemble its fatality records that date back to January, 2000 (pers. comm.); we find FE to be the most comprehensive publicly available database.

Text	Person killed by police?
Alton Sterling was killed by police.	True
Officers shot and killed **Philando Castile**.	True
Officer **Andrew Hanson** was shot.	False
Police report **Megan Short** was fatally shot in apparent murder-suicide.	False

Table 1: Toy examples (with entities in bold) illustrating the problem of extracting from text names of persons who have been killed by police.

and extracting victim names and event details. This approach was recently validated by a Bureau of Justice Statistics study (Banks et al., Dec. 2016) which augmented traditional police-maintained records with media reports, finding twice as many deaths compared to past government analyses. This suggests textual news data has enormous, real value, though *manual* news analysis remains extremely laborious.

We propose to help automate this process by extracting the names of persons killed by police from event descriptions in news articles (Table 1). This can be formulated as either of two cross-document entity-event extraction tasks:

1. Populating an entity-event database: From a corpus of news articles $\mathcal{D}^{(test)}$ over timespan T, extract the names of persons killed by police during that same timespan ($\mathcal{E}^{(pred)}$).

2. Updating an entity-event database: In addition to $\mathcal{D}^{(test)}$, assume access to both a historical database of killings $\mathcal{E}^{(train)}$ and a historical news corpus $\mathcal{D}^{(train)}$ for events that occurred before T. This setting often occurs in practice, and is the focus of this paper; it allows for the use of distantly supervised learn-

ing methods.[2]

The task itself has important social value, but the NLP research community may be interested in a scientific justification as well. We propose that police fatalities are a useful test case for event extraction research. Fatalities are a well defined type of event with clear semantics for coreference, avoiding some of the more complex issues in this area (Hovy et al., 2013). The task also builds on a considerable information extraction literature on knowledge base population (e.g. Craven et al. (1998)). Finally, we posit that the field of natural language processing should, when possible, advance applications of important public interest. Previous work established the value of textual news for this problem, but computational methods could alleviate the scale of manual labor needed to use it.

To introduce this problem, we:

- Define the task of identifying persons killed by police, which is an instance of cross-document entity-event extraction (§3.1).

- Present a new dataset of web news articles collected throughout 2016 that describe possible fatal encounters with police officers (§3.2).

- Introduce, for the database update setting, a distant supervision model (§4) that incorporates feature-based logistic regression and convolutional neural network classifiers under a latent disjunction model.

- Demonstrate the approach's potential usefulness for practitioners: it outperforms two off-the-shelf event extractors (§5) and finds 39 persons not included in the Guardian's "The Counted" database of police fatalities as of January 1, 2017 (§6). This constitutes a promising first step, though performance needs to be improved for real-world usage.

2 Related Work

This task combines elements of information extraction, including: *event extraction* (a.k.a. *semantic parsing*), identifying descriptions of events and their arguments from text, and cross-document *relation extraction*, predicting semantic relations over entities. A fatality event indicates the killing

---

[2]Konovalov et al. (2017) studies the database update task where edits to Wikipedia infoboxes constitute events.

of a particular person; we wish to specifically identify the names of fatality victims mentioned in text. Thus our task could be viewed as unary relation extraction: for a given person mentioned in a corpus, were they killed by a police officer?

Prior work in NLP has produced a number of event extraction systems, trained on text data hand-labeled with a pre-specified ontology, including ones that identify instances of killings (Li and Ji, 2014; Das et al., 2014). Unfortunately, they perform poorly on our task (§5), so we develop a new method.

Since we do not have access to text specifically annotated for police killing events, we instead turn to *distant supervision*—inducing labels by aligning relation-entity entries from a gold standard database to their mentions in a corpus (Craven and Kumlien, 1999; Mintz et al., 2009; Bunescu and Mooney, 2007; Riedel et al., 2010). Similar to this work, Reschke et al. (2014) apply distant supervision to multi-slot, template-based event extraction for airplane crashes; we focus on a simpler unary extraction setting with joint learning of a probabilistic model. Other related work in the cross-document setting has examined joint inference for relations, entities, and events (Yao et al., 2010; Lee et al., 2012; Yang et al., 2015).

Finally, other natural language processing efforts have sought to extract social behavioral event databases from news, such as instances of protests (Hanna, 2017), gun violence (Pavlick et al., 2016), and international relations (Schrodt and Gerner, 1994; Schrodt, 2012; Boschee et al., 2013; O'Connor et al., 2013; Gerrish, 2013). They can also be viewed as event database population tasks, with differing levels of semantic specificity in the definition of "event."

3 Task and Data

3.1 Cross-document entity-event extraction for police fatalties

From a corpus of documents $\mathcal{D}$, the task is to extract a list of candidate person names, $\mathcal{E}$, and for each $e \in \mathcal{E}$ find

$$P(y_e = 1 \mid x_{\mathcal{M}(e)}). \qquad (1)$$

Here $y \in \{0, 1\}$ is the entity-level label where $y_e = 1$ means a person (entity) e was killed by police; $x_{\mathcal{M}(e)}$ are the sentences containing mentions $\mathcal{M}(e)$ of that person. A mention $i \in \mathcal{M}(e)$ is a token span in the corpus. Most entities have

Knowledge base	Historical	Test
FE incident dates	Jan 2000 – Aug 2016	Sep 2016 – Dec 2016
FE gold entities ($\mathcal{G}$)	17,219	452

News dataset	Train	Test
doc. dates	Jan 2016 – Aug 2016	Sep 2016 – Dec 2016
total docs. ($\mathcal{D}$)	866,199	347,160
total ments. ($\mathcal{M}$)	132,833	68,925
pos. ments. ($\mathcal{M}^+$)	11,274	6,132
total entities ($\mathcal{E}$)	49,203	24,550
pos. entities ($\mathcal{E}^+$)	916	258

Table 2: Data statistics for Fatal Encounters (FE) and scraped news documents. $\mathcal{M}$ and $\mathcal{E}$ result from NER processing, while $\mathcal{E}^+$ results from matching textual named entities against the gold-standard database ($\mathcal{G}$).

multiple mentions; a single sentence can contain multiple mentions of different entities.

3.2 News documents

We download a collection of web news articles by continually querying Google News[3] throughout 2016 with lists of police keywords (i.e police, officer, cop etc.) and fatality-related keywords (i.e. kill, shot, murder etc.). The keyword lists were constructed semi-automatically from cosine similarity lookups from the *word2vec* pretrained word embeddings[4] in order to select a high-recall, broad set of keywords. The search is restricted to what Google News defines as a "regional edition" of "United States (English)" which seems to roughly restrict to U.S. news though we anecdotally observed instances of news about events in the U.K. and other countries. We apply a pipeline of text extraction, cleaning, and sentence de-duplication described in the appendix.

3.3 Entity and mention extraction

We process all documents with the open source *spaCy* NLP package[5] to segment sentences, and extract entity mentions. Mentions are token spans that (1) were identified as "persons" by spaCy's named entity recognizer, and (2) have a (firstname, lastname) pair as analyzed by the HAPNIS rule-based name parser,[6] which extracts, for example,

---

	x	z	y
"Hard" training	observed	fixed (distantly labeled)	observed
"Soft" (EM) training	observed	latent	observed
Testing	observed	latent	latent

Table 3: Training and testing settings for mention sentences x, mention labels z, and entity labels y.

(*John*, *Doe*) from the string *Mr. John A. Doe Jr.*.[7]

To prepare sentence text for modeling, our preprocessor collapses the candidate mention span to a special TARGET symbol. To prevent overfitting, other person names are mapped to a different PERSON symbol; e.g. "TARGET was killed in an encounter with police officer PERSON."

There were initially 18,966,757 and 6,061,717 extracted mentions for the train and test periods respectively. To improve precision and computational efficiency, we filtered to sentences that contained at least one police keyword and one fatality keyword. This filter reduced positive entity recall a moderate amount (from 0.68 to 0.57), but removed 99% of the mentions, resulting in the $|\mathcal{M}|$ counts in Table 2.[8]

Other preprocessing steps included heuristics for extraction and name cleanups and are detailed in the appendix.

4 Models

Our goal is to classify entities as to whether they have been killed by police (§4.1). Since we do not have gold-standard labels to train our model, we turn to *distant supervision* (Craven and Kumlien, 1999; Mintz et al., 2009), which heuristically aligns facts in a knowledge base to text in a corpus to impute positive mention-level labels for supervised learning. Previous work typically examines distant supervision in the context of binary relation extraction (Bunescu and Mooney, 2007; Riedel et al., 2010; Hoffmann et al., 2011), but we are concerned with the unary predicate "person was killed by police." As our gold standard knowledge

---

base ($\mathcal{G}$), we use Fatal Encounters' (FE) publicly available dataset: around 18,000 entries of victim's name, age, gender and race as well as location, cause and date of death. (We use a version of the FE database downloaded Feb. 27, 2017.) We compare two different distant supervision training paradigms (Table 3): "hard" label training (§4.2) and "soft" EM-based training (§4.3). This section also details mention-level models (§4.4,§4.5) and evaluation (§4.6).

4.1 Approach: Latent disjunction model

Our discriminative model is built on mention-level probabilistic classifiers. Recall a single entity will have one or more mentions (i.e. the same name occurs in multiple sentences in our corpus). For a given mention i in sentence x_i, our model predicts whether the person is described as having been killed by police, $z_i = 1$, with a binary logistic model,

$$P(z_i = 1 \mid x_i) = \sigma(\beta^{\mathsf{T}} f_\gamma(x_i)). \quad (2)$$

We experiment with both logistic regression (§4.4) and convolutional neural networks (§4.5) for this component, which use logistic regression weights β and feature extractor parameters γ. Then we must somehow aggregate mention-level decisions to determine entity labels y_e.[9] If a human reader were to observe at least one sentence that states a person was killed by police, they would infer that person was killed by police. Therefore we aggregate an entity's mention-level labels with a deterministic disjunction:

$$P(y_e = 1 \mid z_{\mathcal{M}(e)}) = 1 \left\{ \vee_{i \in \mathcal{M}(e)} z_i \right\}. \quad (3)$$

At test time, z_i is latent. Therefore the correct inference for an entity is to marginalize out the model's uncertainty over z_i:

$$P(y_e = 1 | x_{\mathcal{M}(e)}) = 1 - P(y_e = 0 | x_{\mathcal{M}(e)}) \quad (4)$$

$$= 1 - P(z_{\mathcal{M}(e)} = \vec{0} \mid x_{\mathcal{M}(e)}) \quad (5)$$

$$= 1 - \prod_{i \in \mathcal{M}(e)} (1 - P(z_i = 1 \mid x_i)). \quad (6)$$

Eq. 6 is the *noisyor* formula (Pearl, 1988; Craven and Kumlien, 1999). Procedurally, it counts strong probabilistic predictions as evidence, but can also

incorporate a large number of weaker signals as positive evidence as well.[10]

In order to train these classifiers, we need mention-level labels (z_i) which we impute via two different distant supervision labeling methods: "hard" and "soft."

4.2 "Hard" distant label training

In "hard" distant labeling, labels for mentions in the training data are heuristically imputed and directly used for training. We use two labeling rules. First, **name-only:**

$$z_i = 1 \text{ if } \exists e \in \mathcal{G}^{(train)} : \text{name}(i) = \text{name}(e). \quad (7)$$

This is the direct unary predicate analogue of Mintz et al. (2009)'s *distant supervision assumption*, which assumes every mention of a gold-positive entity exhibits a description of a police killing.

This assumption is not correct. We manually analyze a sample of positive mentions and find 36 out of 100 name-only sentences did not express a police fatality event—for example, sentences contain commentary, or describe killings not by police. This is similar to the precision for distant supervision of binary relations found by Riedel et al. (2010), who reported 10–38% of sentences did not express the relation in question.

Our higher precision rule, **name-and-location**, leverages the fact that the location of the fatality is also in the Fatal Encounters database and requires both to be present:

$$z_i = 1 \text{ if } \exists e \in \mathcal{G}^{(train)} :$$
$$\text{name}(i) = \text{name}(e) \text{ and location}(e) \in x_i. \quad (8)$$

We use this rule for training since precision is slightly better, although there is still a considerable level of noise.

4.3 "Soft" (EM) joint training

At training time, the *distant supervision assumption* used in "hard" label training is flawed: many positively-labeled mentions are in sentences that

---

[9]An alternative approach is to aggregate features across mentions into an entity-level feature vector (Mintz et al., 2009; Riedel et al., 2010); but here we opt to directly model at the mention level, which can use contextual information.

[10]In early experiments, we experimented with other, more ad-hoc aggregation rules with a "hard"-trained model. The maximum and arithmetic mean functions performed worse than *noisyor*, giving credence to the disjunction model. The sum rule ($\sum_i P(z_i = 1 \mid x_i)$) had similar ranking performance as *noisyor*—perhaps because it too can use weak signals, unlike mean or max—though it does not yield proper probabilities between 0 and 1.

do not assert the person was killed by a police officer. Alternatively, at training time we can treat z_i as a latent variable and assume, as our model states, that *at least one* of the mentions asserts the fatality event, but leave uncertainty over which mention (or multiple mentions) conveys this information. This corresponds to multiple instance learning (MIL; Dietterich et al. (1997)) which has been applied to distantly supervised relation extraction by enforcing the *at least one* constraint at training time (Bunescu and Mooney, 2007; Riedel et al., 2010; Hoffmann et al., 2011; Surdeanu et al., 2012; Ritter et al., 2013). Our approach differs by using exact marginal posterior inference for the E-step.

With z_i as latent, the model can be trained with the EM algorithm (Dempster et al., 1977). We initialize the model by training on the "hard" distant labels (§4.2), and then learn improved parameters by alternating E- and M-steps.

The **E-step** requires calculating the marginal posterior probability for each z_i,

$$q(z_i) := P(z_i \mid x_{\mathcal{M}(e_i)}, y_{e_i}). \qquad (9)$$

This corresponds to calculating the posterior probability of a disjunct, given knowledge of the output of the disjunction, and prior probabilities of all disjuncts (given by the mention-level classifier).

Since $P(z \mid x, y) = P(z, y \mid x)/P(y \mid x)$,

$$q(z_i = 1) = \frac{P(z_i = 1, y_{e_i} = 1 | x_{\mathcal{M}(e_i)})}{P(y_{e_i} = 1 | x_{\mathcal{M}(e_i)})}. \qquad (10)$$

The numerator simplifies to the mention prediction $P(z_i = 1 \mid x_i)$ and the denominator is the entity-level *noisyor* probability (Eq. 6). This has the effect of taking the classifier's predicted probability and increasing it slightly (since Eq. 10's denominator is no greater than 1); thus the disjunction constraint implies a soft positive labeling. In the case of a negative entity with $y_e = 0$, the disjunction constraint implies all $z_{\mathcal{M}(e)}$ stay clamped to 0 as in the "hard" label training method.

The $q(z_i)$ posterior weights are then used for the **M-step**'s expected log-likelihood objective:

$$\max_\theta \sum_i \sum_{z \in \{0,1\}} q(z_i = z) \log P_\theta(z_i = z \mid x_i). \qquad (11)$$

This objective (plus regularization) is maximized with gradient ascent as before.

This approach can be applied to any mention-level probabilistic model; we explore two in the next sections.
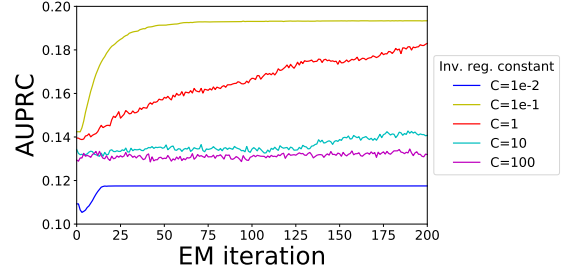

Figure 1: For soft-LR (EM), area under precision recall curve (AUPRC) results on the test set during training, for different inverse regularization values (C, the parameters' prior variance).

	Features
$D1$	length 3 dependency paths that include TARGET: word, POS, dep. label
$D2$	length 3 dependency paths that include TARGET: word and dep. label
$D3$	length 3 dependency paths that include TARGET: word and POS
$D4$	all length 2 dependency paths with word, POS, dep. labels
$N1$	n-grams length 1, 2, 3
$N2$	n-grams length 1, 2, 3 plus POS tags
$N3$	n-grams length 1, 2, 3 plus directionality and position from TARGET
$N4$	concatenated POS tags of 5-word window centered on TARGET
$N5$	word and POS tags for 5-word window centered on TARGET

Table 4: Feature templates for logistic regression grouped into syntactic dependencies (D) and N-gram (N) features.

4.4 Feature-based logistic regression

We construct hand-crafted features for regularized logistic regression (LR) (Table 4), designed to be broadly similar to the n-gram and syntactic dependency features used in previous work on feature-based semantic parsing (e.g. Das et al. (2014); Thomson et al. (2014)). We use randomized feature hashing (Weinberger et al., 2009) to efficiently represent features in 450,000 dimensions, which achieved similar performance as an explicit feature representation. The logistic regression weights (β in Eq. 2) are learned with *scikit-learn* (Pedregosa et al., 2011).[11] For EM (soft-LR) training, the test set's area under the precision recall curve converges after 96 iterations (Fig. 1).

---

[11]With *FeatureHasher*, L2 regularization, 'lbfgs' solver, and inverse strength $C = 0.1$, tuned on a development dataset in "hard" training; for EM training the same regularization strength performs best.

4.5 Convolutional neural network

We also train a convolutional neural network (CNN) classifier, which uses word embeddings and their nonlinear compositions to potentially generalize better than sparse lexical and n-gram features. CNNs have been shown useful for sentence-level classification tasks (Kim, 2014; Zhang and Wallace, 2015), relation classification (Zeng et al., 2014) and, similar to this setting, event detection (Nguyen and Grishman, 2015). We use Kim (2014)'s open-source CNN implementation,[12] where a logistic function makes the final mention prediction based on max-pooled values from convolutional layers of three different filter sizes, whose parameters are learned (γ in Eq. 2). We use pretrained word embeddings for initialization,[13] and update them during training. We also add two special vectors for the TARGET and PERSON symbols, initialized randomly.[14]

For training, we perform stochastic gradient descent for the negative expected log-likelihood (Eq. 11) by sampling with replacement fifty mention-label pairs for each minibatch, choosing each $(i, k) \in \mathcal{M} \times \{0, 1\}$ with probability proportional to $q(z_i = k)$. This strategy attains the same expected gradient as the overall objective. We use "epoch" to refer to training on 265,700 examples (approx. twice the number of mentions). Unlike EM for logistic regression, we do not run gradient descent to convergence, instead applying an E-step every two epochs to update q; this approach is related to incremental and online variants of EM (Neal and Hinton, 1998; Liang and Klein, 2009), and is justified since both SGD and E-steps improve the evidence lower bound (ELBO). It is also similar to Salakhutdinov et al. (2003)'s expectation gradient method; their analysis implies the gradient calculated immediately after an E-step is in fact the gradient for the marginal log-likelihood. We are not aware of recent work that uses EM to train latent-variable neural network models, though this combination has been explored (e.g. Jordan and Jacobs (1994))

4.6 Evaluation

On documents from the test period (Sept–Dec 2016), our models predict entity-level labels

Figure 2: At test time, there are matches between the knowledge base and the news reports both for persons killed during the test period ("positive") and persons killed before it ("historical"). Historical cases are excluded from evaluation.
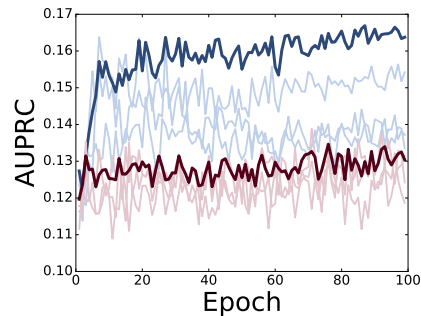
Figure 3: Test set AUPRC for three runs of soft-CNN (EM) (blue, higher in graph), and hard-CNN (red, lower in graph). Darker lines show performance of averaged predictions.

$P(y_e = 1 \mid x_{\mathcal{M}(e)})$ (Eq. 6), and we wish to evaluate whether retrieved entities are listed in Fatal Encounters as being killed during Sept–Dec 2016. We rank entities by predicted probabilities to construct a precision-recall curve (Fig. 4, Table 5). Area under the precision-recall curve (AUPRC) is calculated with a trapezoidal rule; F1 scores are shown for convenient comparison to non-ranking approaches (§5).

Excluding historical fatalities: Our model gives strong positive predictions for many people who were killed by police before the test period (i.e. before Sept 2016), when news articles contain discussion of historical police killings. We exclude these entities from evaluation, since we want to simulate an update to a fatality database (Fig 2). Our test dataset contains 1,148 such historical entities.

Data upper bound: Of the 452 gold entities in the FE database at test time, our news corpus only contained 258 (Table 2), hence the data up-

---

[12]https://github.com/yoonkim/CNN_sentence

[13]From the same *word2vec* embeddings used in §3.

[14]Training proceeds with ADADELTA (Zeiler, 2012). We tested several different settings of dropout and L2 regularization hyperparameters on a development set, but found mixed results, so used their default values.
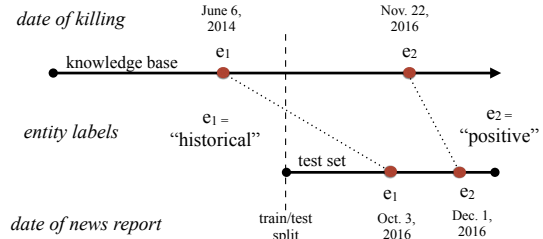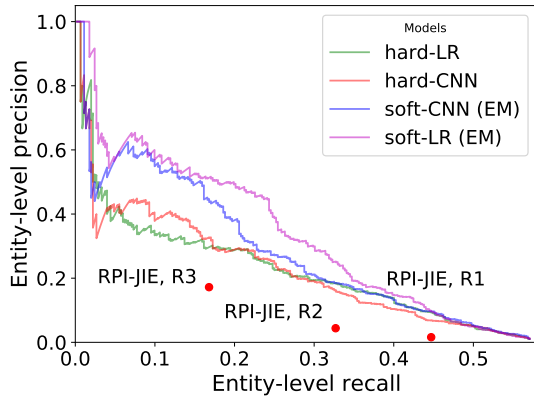
Figure 4: Precision-recall curves for the given models.

Model	AUPRC	F1
hard-LR, dep. feats.	0.117	0.229
hard-LR, n-gram feats.	0.134	0.257
hard-LR, all feats.	0.142	0.266
hard-CNN	0.130	0.252
soft-CNN (EM)	0.164	0.267
soft-LR (EM)	**0.193**	**0.316**
Data upper bound (§4.6)	0.57	0.73

Table 5: Area under precision-recall curve (AUPRC) and F1 (its maximum value from the PR curve) for entity prediction on the test set.

per bound of 0.57 recall, which also gives an upper bound of 0.57 on AUPRC. This is mostly a limitation of our news corpus; though we collect hundreds of thousands of news articles, it turns out Google News only accesses a subset of relevant web news, as opposed to more comprehensive data sources manually reviewed by Fatal Encounters' human experts. We still believe our dataset is large enough to be realistic for developing better methods, and expect the same approaches could be applied to a more comprehensive news corpus.

5 Off-the-shelf event extraction baselines

From a practitioner's perspective, a natural first approach to this task would be to run the corpus of police fatality documents through pre-trained, "off-the-shelf" event extractor systems that could identify killing events. In modern NLP research, a major paradigm for event extraction is to formulate a hand-crafted ontology of event classes, annotate a small corpus, and craft supervised learning systems to predict event parses of documents.

	Rule	Prec.	Recall	F1
SEMAFOR	R1	0.011	0.436	0.022
	R2	0.031	0.162	0.051
	R3	0.098	0.009	0.016
RPI-JIE	R1	0.016	0.447	0.030
	R2	0.044	0.327	0.078
	R3	0.172	0.168	**0.170**
Data upper bound (§4.6)		1.0	0.57	0.73

Table 6: Precision, recall, and F1 scores for test data using event extractors SEMAFOR and RPI-JIE and rules R1-R3 described below.

We evaluate two freely available, off-the-shelf event extractors that were developed under this paradigm: SEMAFOR (Das et al., 2014), and the RPI Joint Information Extraction System (RPI-JIE) (Li and Ji, 2014), which output semantic structures following the FrameNet (Fillmore et al., 2003) and ACE (Doddington et al., 2004) event ontologies, respectively.[15] Pavlick et al. (2016) use RPI-JIE to identify instances of gun violence.

For each mention $i \in \mathcal{M}$ we use SEMAFOR and RPI-JIE to extract event tuples of the form $t_i =$ (event type, agent, patient) from the sentence x_i. We want the system to detect (1) killing events, where (2) the killed person is the target mention i, and (3) the person who killed them is a police officer. We implement a small progression of these neo-Davidsonian (Parsons, 1990) conjuncts with rules to classify $z_i = 1$ if:[16]

- **(R1)** the event type is 'kill.'
- **(R2)** R1 holds and the patient token span contains e_i.

---

[15] Many other annotated datasets encode similar event structures in text, but with lighter ontologies where event classes directly correspond with lexical items—including PropBank, Prague Treebank, DELPHI-IN MRS, and Abstract Meaning Representation (Kingsbury and Palmer, 2002; Hajic et al., 2012; Oepen et al., 2014; Banarescu et al., 2013). We assume such systems are too narrow for our purposes, since we need an extraction system to handle different trigger constructions like "killed" versus "shot dead."

[16] For SEMAFOR, we use the FrameNet 'Killing' frame with frame elements 'Victim' and 'Killer'. For RPI-JIE, we use the ACE 'life/die' event type/subtype with roles 'victim' and 'agent'. SEMAFOR defines a token span for every argument; RPI-JIE/ACE defines two spans, both a head word and entity extent; we use the entity extent. SEMAFOR only predicts spans as event arguments, while RPI-JIE also predicts entities as event arguments, where each entity has a within-text coreference chain over one or more mentions; since we only use single sentences, these chains tend to be small, though they do sometimes resolve pronouns. For determining R2 and R3, we allow a match on any of an entity's extents from any of its mentions.

- **(R3)** R2 holds and the agent token span contains a police keyword.

As in §4.1 (Eq. 3), we aggregate mention-level z_i predictions to obtain entity-level predictions with a deterministic OR of $z_{\mathcal{M}(e)}$.

RPI-JIE under the full R3 system performs best, though all results are relatively poor (Table 6). Part of this is due to inherent difficulty of the task, though our task-specific model still outperforms (Table 5). We suspect a major issue is that these systems heavily rely on their annotated training sets and may have significant performance loss on new domains, or messy text extracted from web news, suggesting domain transfer for future work.

6 Results and discussion

Significance testing: We would like to test robustness of performance results to the finite datasets with bootstrap testing (Berg-Kirkpatrick et al., 2012), which can accomodate performance metrics like AUPRC. It is not clear what the appropriate unit of resampling should be—for example, parsing and machine translation research in NLP often resamples sentences, which is inappropriate for our setting. We elect to resample documents in the test set, simulating variability in the generation and retrieval of news articles. Standard errors for one model's AUPRC and F1 are in the range 0.004–0.008 and 0.008–0.010 respectively; we also note pairwise significance test results. See appendix for details.

Overall performance: Our results indicate our model is better than existing computational methods methods to extract names of people killed by police, by comparing to F1 scores of off-the-shelf extractors (Table 5 vs. Table 6; differences are statistically significant).

We also compare entities extracted from our test dataset to the Guardian's "The Counted" database of U.S. police killings during the span of the test period (Sept.–Dec., 2016),[17] and found 39 persons they did not include in the database, but who were in fact killed by police. This implies our approach could augment journalistic collection efforts. Additionally, our model could help practitioners by presenting them with sentence-level information in the form of Table 7; we hope this could decrease the amount of time and emotional toll required to maintain real-time updates of police fatality databases.

[17]https://www.theguardian.com/us-news/series/counted-us-police-killings, downloaded Jan. 1, 2017.

CNN: Model predictions were relatively unstable during the training process. Despite the fact that EM's evidence lower bound objective $(H(Q) + E_Q[\log P(Z, Y|X)])$ converged fairly well on the training set, test set AUPRC substantially fluctuated as much as 2% between epochs, and also between three different random initializations for training (Fig. 3). We conducted these multiple runs initially to check for variability, then used them to construct a basic ensemble: we averaged the three models' mention-level predictions before applying *noisyor* aggregation. This outperformed the individual models—especially for EM training—and showed less fluctuation in AUPRC, which made it easier to detect convergence. Reported performance numbers in Table 5 are with the average of all three runs from the final epoch of training.

LR vs. CNN: After feature ablation we found that hard-CNN and hard-LR with n-gram features (N1-N5) had comparable AUPRC values (Table 5). But adding dependency features (D1-D4) caused the logistic regression models to outperform the neural networks (albeit with bare significance: $p = 0.046$). We hypothesize these dependency features capture longer-distance semantic relationships between the entity, fatality trigger word, and police officer, which short n-grams cannot. Moving to sequence or graph LSTMs may better capture such dependencies.

Soft (EM) training: Using the EM algorithm gives substantially better performance: for the CNN, AUC improves from 0.130 to 0.164, and for LR, from 0.142 to 0.193. (Both improvements are statistically significant.) Logistic regression with EM training is the most accurate model. Examining the precision-recall curves (Fig. 4), many of the gains are in the higher confidence predictions (left side of figure). In fact, the soft EM model makes fewer strongly positive predictions: for example, hard-LR predicts $y_e = 1$ with more than 99% confidence for 170 out of 24,550 test set entities, but soft-LR does so for only 24. This makes sense given that the hard-LR model at training time assumes that many more positive entity mentions are evidence of a killing than they are in reality (§4.2).

Manual analysis: Manual analysis of false positives indicates misspellings or mismatches of names, police fatalities outside of the U.S., people who were shot by police but not killed, and names of police officers who were killed are com-

entity (e)	ment.(i) prob.	ment. text (x_i)
Keith Scott (true pos)	0.98	Charlotte protests Charlotte's Mayor Jennifer Roberts speaks to reporters the morning after protests against the police shooting of **Keith Scott**, in Charlotte, North Carolina .
Terence Crutcher (true pos)	0.96	Tulsa Police Department released video footage Monday, Sept. 19, 2016, showing white Tulsa police officer Betty Shelby fatally shooting **Terence Crutcher,** 40, a black man police later determined was unarmed.
Mark Duggan (false pos)	0.97	The fatal shooting of **Mark Duggan** by police led to some of the worst riots in England's recent history.
Logan Clarke (false pos)	0.92	**Logan Clarke** was shot by a campus police officer after waving kitchen knives at fellow students outside the cafeteria at Hug High School in Reno, Nevada, on December 7.

Table 7: Example of highly ranked entities, with selected mention predictions and text.

mon false positive errors (see detailed table in the appendix). This suggests many prediction errors are from ambiguous or challenging cases.[18]

Future work: While we have made progress on this application, more work is necessary for accuracy to be high enough to be useful for practitioners. Our model allows for the use of mention-level semantic parsing models; systems with explicit trigger/agent/patient representations, more like traditional event extraction systems, may be useful, as would more sophisticated neural network models, or attention models as an alternative to disjunction aggregation (Lin et al., 2016).

One goal is to use our model as part of a semi-automatic system, where people manually review a ranked list of entity suggestions. In this case, it is more important to focus on improving recall—specifically, improving precision at high-recall points on the precision-recall curve. Our best models, by contrast, tend to improve precision at lower-recall points on the curve. Higher recall may be possible through cost-sensitive training (e.g. Gimpel and Smith (2010)) and using features from beyond single sentences within the document.

Furthermore, our dataset could be used to contribute to communication studies, by exploring research questions about the dynamics of media attention (for example, the effect of race and geography on coverage of police killings), and discussions of historical killings in news—for example, many articles in 2016 discussed Michael Brown's 2014 death in Ferguson, Missouri. Improving NLP analysis of historical events would also be useful for the event extraction task itself, by delineating between recent events that re-

quire a database update, versus historical events that appear as "noise" from the perspective of the database update task. Finally, it may also be possible to adapt our model to extract other types of social behavior events.

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*. Association for Computational Linguistics, Sofia, Bulgaria, pages 178–186. http://www.aclweb.org/anthology/W13-2322.

Duren Banks, Paul Ruddle, Erin Kennedy, and Michael G. Planty. 2016. Arrest-related deaths program redesign study, 2015–16: Preliminary findings. Technical report, Technical Report NCJ 250112.

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in NLP. In *Proceedings of EMNLP*.

Elizabeth Boschee, Premkumar Natarajan, and Ralph Weischedel. 2013. Automatic extraction of events from open source text for predictive forecasting. *Handbook of Computational Approaches to Counterterrorism* page 51.

Razvan Bunescu and Raymond Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of*

---

[18]We attempted to correct non-U.S. false positive errors by using CLAVIN, an open-source country identifier, but this significantly hurt recall.

ACL. Association for Computational Linguistics, Prague, Czech Republic, pages 576–583. http://www.aclweb.org/anthology/P07-1073.

Mark Craven and Johan Kumlien. 1999. Constructing biological knowledge bases by extracting information from text sources. In *ISMB*. pages 77–86.

Mark Craven, Andrew McCallum, Dan PiPasquo, Tom Mitchell, and Dayne Freitag. 1998. Learning to extract symbolic knowledge from the World Wide Web. In *Proceedings of AAAI*.

Dipanjan Das, Desai Chen, Andre F. T. Martins, Nathan Schneider, and Noah A. Smith. 2014. Frame-semantic parsing. *Computational Linguistics*.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (methodological)* pages 1–38.

Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence* 89(1):31–71.

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 *LREC*. volume 2, page 1.

Charles J. Fillmore, Christopher R. Johnson, and Miriam R.L. Petruck. 2003. Background to FrameNet. *International Journal of Lexicography*.

Sean M Gerrish. 2013. *Applications of Latent Variable Models in Modeling Influence and Decision Making*. Ph.D. thesis, Princeton University.

Kevin Gimpel and Noah A. Smith. 2010. Softmax-margin CRFs: Training log-linear models with cost functions. In *Proceedings of NAACL-HLT*. Association for Computational Linguistics, pages 733–736.

Jan Hajic, Eva Hajicová, Jarmila Panevová, Petr Sgall, Ondrej Bojar, Silvie Cinková, Eva Fucíková, Marie Mikulová, Petr Pajas, Jan Popelka, et al. 2012. Announcing prague czech-english dependency treebank 2.0. In *LREC*. pages 3153–3160.

Alex Hanna. 2017. MPEDS: Automating the generation of protest event data. *SocArXiv*.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of ACL*. Association for Computational Linguistics, Portland, Oregon, USA, pages 541–550. http://www.aclweb.org/anthology/P11-1055.

Eduard Hovy, Teruko Mitamura, Felisa Verdejo, Jun Araki, and Andrew Philpot. 2013. Events are not simple: Identity, non-identity, and quasi-identity. In *Workshop on Events: Definition, Detection, Coreference, and Representation*. Association for Computational Linguistics, Atlanta, Georgia, pages 21–28. http://www.aclweb.org/anthology/W13-1203.

Michael I Jordan and Robert A Jacobs. 1994. Hierarchical mixtures of experts and the em algorithm. *Neural computation* 6(2):181–214.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*.

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In *LREC*. pages 1989–1993.

Alexander Konovalov, Benjamin Strauss, Alan Ritter, and Brendan O'Connor. 2017. Learning to extract events from knowledge base revisions. In *Proceedings of WWW*.

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

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of ACL*.

Percy Liang and Dan Klein. 2009. Online EM for unsupervised models. In *Proceedings of NAACL*. Boulder, Colorado. http://www.aclweb.org/anthology/N/N09/N09-1069.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of ACL*. Association for Computational Linguistics, Berlin, Germany, pages 2124–2133. http://www.aclweb.org/anthology/P16-1200.

Kristian Lum and Patrick Ball. 2015. Estimating undocumented homicides with two lists and list dependence. *Human Rights Data Analysis Group* https://hrdag.org/wp-content/uploads/2015/07/2015-hrdag-estimating-undoc-homicides.pdf.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL*. Suntec, Singapore. http://www.aclweb.org/anthology/P/P09/P09-1113.

Radford M Neal and Geoffrey E Hinton. 1998. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, Springer, pages 355–368.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of ACL*.

Brendan O'Connor, Brandon Stewart, and Noah A. Smith. 2013. Learning to extract international relations from political context. In *Proceedings of ACL*.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajic, Angelina Ivanova, and Yi Zhang. 2014. Semeval 2014 task 8: Broad-coverage semantic dependency parsing. In *Proceedings of SemEval*. http://www.aclweb.org/anthology/S14-2008.

Terence Parsons. 1990. *Events in the Semantics of English*. Cambridge, MA: MIT Press.

Ellie Pavlick, Heng Ji, Xiaoman Pan, and Chris Callison-Burch. 2016. The Gun Violence Database: A new task and data set for NLP. In *Proceedings of EMNLP*. https://aclweb.org/anthology/D16-1106.

Judea Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Kevin Reschke, Martin Jankowiak, Mihai Surdeanu, Christopher D. Manning, and Daniel Jurafsky. 2014. Event extraction using distant supervision. In *Language Resources and Evaluation Conference (LREC)*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pages 148–163.

Alan Ritter, Luke Zettlemoyer, Oren Etzioni, et al. 2013. Modeling missing data in distant supervision for information extraction. *TACL* .

Ruslan Salakhutdinov, Sam T Roweis, and Zoubin Ghahramani. 2003. Optimization with EM and expectation-conjugate-gradient. In *Proceedings of ICML*.

Philip A. Schrodt. 2012. Precedents, progress, and prospects in political event data. *International Interactions* 38(4):546–569.

Philip A. Schrodt and Deborah J. Gerner. 1994. Validity assessment of a machine-coded event data set for the Middle East, 1982-1992. *American Journal of Political Science* .

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of EMNLP*. http://www.aclweb.org/anthology/D12-1042.

Sam Thomson, Brendan O'Connor, Jeffrey Flanigan, David Bamman, Jesse Dodge, Swabha Swayamdipta, Nathan Schneider, Chris Dyer, and Noah A. Smith. 2014. CMU: Arc-factored, discriminative semantic dependency parsing. In *Proceedings of SemEval*. http://www.aclweb.org/anthology/S14-2027.

Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. 2009. Feature hashing for large scale multitask learning. In *Proceedings of ICML*.

Bishan Yang, Claire Cardie, and Peter Frazier. 2015. A hierarchical distance-dependent Bayesian model for event coreference resolution. *TACL* 3.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2010. Collective cross-document relation extractionwithout labelled data. In *Proceedings of EMNLP*.

Matthew D. Zeiler. 2012. Adadelta: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701* .

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING*.

Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820* .