Event Schema Induction with a Probabilistic Entity-Driven Model

Nathanael Chambers
United States Naval Academy
Annapolis, MD 21402
nchamber@usna.edu

Abstract

Event schema induction is the task of learning high-level representations of complex events (e.g., a *bombing*) and their entity roles (e.g., *perpetrator* and *victim*) from unlabeled text. Event schemas have important connections to early NLP research on frames and scripts, as well as modern applications like template extraction. Recent research suggests event schemas can be learned from raw text. Inspired by a pipelined learner based on named entity coreference, this paper presents the first generative model for schema induction that integrates coreference chains into learning. Our generative model is conceptually simpler than the pipelined approach and requires far less training data. It also provides an interesting contrast with a recent HMM-based model. We evaluate on a common dataset for template schema extraction. Our generative model matches the pipeline's performance, and outperforms the HMM by 7 F1 points (20%).

1 Introduction

Early research in language understanding focused on high-level semantic representations to drive their models. Many proposals, such as frames and scripts, used rich event schemas to model the situations described in text. While the field has since focused on more shallow approaches, recent work on schema induction shows that event schemas might be learnable from raw text. This paper continues the trend, addressing the question, can event schemas be induced from raw text without prior knowledge? We present a new generative model for event schemas, and it produces state-of-the-art induction results, including a 7 F1 point gain over a different generative proposal developed in parallel with this work.

Event schemas are unique from most work in information extraction (IE). Current relation discovery (Banko et al., 2007a; Carlson et al., 2010b) focuses on atomic facts and relations. Event schemas build relations into coherent event structures, often called *templates* in IE. For instance, an election template jointly connects that *obama won a presidential election* with *romney was the defeated*, *the election occurred in 2012*, and *the popular vote was 50-48*. The entities in these relations fill specific semantic roles, as in this template schema:

Template Schema for Elections
(*events*: nominate, vote, elect, win, declare, concede)

Date:	Timestamp
Winner:	Person
Loser:	Person
Position:	Occupation
Vote:	Number

Traditionally, template extractors assume foreknowledge of the event schemas. They know a Winner exists, and research focuses on supervised learning to extract winners from text. This paper focuses on the other side of the supervision spectrum. The learner receives no human input, and it first induces a schema before extracting instances of it.

Our proposed model contributes to a growing line of research in schema induction. The majority of previous work relies on ad-hoc clustering algorithms (Filatova et al., 2006; Sekine, 2006; Chambers and Jurafsky, 2011). Chambers and Jurafsky is a pipelined approach, learning events first, and later learning syntactic patterns as fillers. It requires

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1797–1807,
Seattle, Washington, USA, 18-21 October 2013.

several ad-hoc metrics and parameters, and it lacks the benefits of a formal model. However, central to their algorithm is the use of coreferring entity mentions to knit events and entities together into an event schema. We adapt this entity-driven approach to a single model that requires fewer parameters and far less training data. Further, experiments show state-of-the-art performance.

Other research conducted at the time of this paper also proposes a generative model for schema induction (Cheung et al., 2013). Theirs is not entity-based, but instead uses a sequence model (HMM-based) of verb clauses. These two papers thus provide a unique opportunity to compare two very different views of document structure. One is entity-driven, modeling an entity's role by its coreference chain. The other is clause-driven, classifying individual clauses based on text sequence. Each model makes unique assumptions, providing an interesting contrast. Our entity model outperforms by 7 F1 points on a common extraction task.

The rest of the paper describes in detail our main contributions: (1) the first *entity-based* generative model for schema induction, (2) a direct pipeline/formal model comparison, (3) results improving state-of-the-art performance by 20%, and (4) schema induction from the smallest amount of training data to date.

2 Previous Work

Unsupervised learning for information extraction usually learns binary relations and atomic facts. Models can learn relations like *Person is married to Person* without labeled data (Banko et al., 2007b), or rely on seed examples for ontology induction (*dog is a mammal*) and attribute extraction (*dogs have tails*) (Carlson et al., 2010b; Carlson et al., 2010a; Huang and Riloff, 2010; Durme and Pasca, 2008). These do not typically capture the deeper connections modeled by event schemas.

Algorithms that do focus on event schema extraction typically require both the schemas and labeled corpora, such as rule-based approaches (Chinchor et al., 1993; Rau et al., 1992) and modern supervised classifiers (Freitag, 1998; Chieu et al., 2003; Bunescu and Mooney, 2004; Patwardhan and Riloff, 2009; Huang and Riloff, 2011). Classifiers rely on the labeled examples' surrounding context for features (Maslennikov and Chua, 2007). Weakly supervised learning removes some of the need for labeled data, but most still require the event schemas. One common approach is to begin with unlabeled, but clustered event-specific documents, and extract common word patterns as extractors (Riloff and Schmelzenbach, 1998; Sudo et al., 2003; Riloff et al., 2005; Filatova et al., 2006; Patwardhan and Riloff, 2007; Chen et al., 2011). Bootstrapping with seed examples of known slot fillers has been shown to be effective (Yangarber et al., 2000; Surdeanu et al., 2006).

Shinyama and Sekine (2006) presented *unrestricted relation discovery* to discover relations in unlabeled documents. Their algorithm used redundant documents (e.g., all describe Hurricane Ivan) to observe repeated proper nouns. The approach requires many documents about the exact same event instance, and relations are binary (not schemas) over repeated named entities. Our model instead learns *schemas* from documents with mixed topics that don't describe the same event, so repeated proper nouns are less helpful.

Chen et al. (2011) perform relation extraction with no supervision on earthquake and finance domains. Theirs is a generative model that represents relations as predicate/argument pairs. As with others, training data is pre-clustered by event type and there is no schema connection between relations.

This paper builds the most on Chambers and Jurafsky (2011). They learned event schemas with a three-stage clustering algorithm that included a requirement to retrieve extra training data. This paper removes many of these complexities. We present a formal model that uniquely models coreference chains. Advantages include a *joint* clustering of events and entities, and a formal probabilistic interpretation of the resulting schemas. We achieve better performance, and do so with far less training data.

Cheung et al. (2013) is most related as a generative formulation of schema induction. They propose an HMM-based model over latent event variables, where each variable generates the observed clauses. Latent schema variables generate the event variables (in the spirit of preliminary work by O'Connor (2012)). There is no notion of an entity, so learning uses text mentions and relies on the local HMM win-

message: id	dev-muc3-0112 (bellcore, mitre)
incident: date	10 mar 89
incident: location	peru: huanuco, ambo (town)
incident: type	bombing
incident: stage	accomplished
incident: instrument	explosive: "-"
perp: individual	"shining path members"
perp: organization	"shining path"

Figure 1: A subset of the slots in a MUC-4 template.

dow for event transitions. Their model was created in parallel with our work, and provides a nice contrast in both approach and results. Ours outperforms their model by 20% on a MUC-4 evaluation.

In summary, this paper extends most previous work on event schema induction by removing the supervision. Of the recent 'unsupervised' work, we present the first entity-driven generative model, and we experiment on a mixed-domain corpus.

3 Dataset: The MUC-4 Corpus

The corpus from the Message Understanding Conference (MUC-4) serves as the challenge text (Sundheim, 1991), and will ground discussion of our model. MUC-4 is also used by the closest previous work. It contains Latin American newswire about terrorism events, and it provides a set of hand-constructed event schemas that are traditionally called template schemas. It also maps labeled templates to the text, providing a dataset for template extraction evaluations. Until very recently, only extraction has been evaluated. We too evaluate our model through extraction, but we also compare our learned schemas to the hand-created template schemas. An example of a filled in MUC-4 template is given in Figure 1.

The MUC-4 corpus defines six template types: **Attack**, **Kidnapping**, **Bombing**, **Arson**, **Robbery**, and **Forced Work Stoppage**. Documents are often labeled with more than one template and type. Many include multiple events at different times in different locations. The corpus is particularly challenging because template schemas are inter-mixed and entities can play multiple roles across instances.

The training corpus contains 1300 documents, 733 of which are labeled with at least one schema. 567 documents are not labeled with any schemas. These unlabeled documents are articles that report on non-specific political events and speeches. They make the corpus particularly challenging. The development and test sets each contain 200 documents.

4 A Generative Model for Event Schemas

This paper's model is an entity-based approach, similar in motivation to Haghighi and Klein (2010) and the pipelined induction of Chambers and Jurafsky (2011). Coreference resolution guides the learning by providing a set of pre-resolved entities. Each entity receives a schema role label, so it allows *all mentions* of the entity to inform that role choice. This important constraint links coreferring mentions to the same schema role, and distinguishes our approach from others (Cheung et al., 2013).

4.1 Illustration

The model represents a document as a set of entities. An entity is a set of entity mentions clustered by coreference resolution. We will use the following two sentences for illustration:

> *A truck bomb exploded near the embassy.*
> *Three militia planted it, and then they fled.*

This text contains five entity mentions. A perfect coreference resolution system will resolve these five mentions into three entities:

Entity Mentions	**Entities**	**Roles**
a truck bomb	(a truck bomb, it)	Instrument
the embassy	(the embassy)	Target
three militia	(three militia, they)	Perpetrator
it		
they		

The schema roles, or template *slots*, are the type of target knowledge we want to learn. Each entity will be labeled with both a slot variable s and a template variable t (e.g., the *s=perpetrator* of a *t=bombing*). The lexical context of the entity mentions guides the learning model to this end.

4.2 Definitions

A document $d \in D$ is represented as a set of entities E_d. Each entity $e \in E_d$ is a triple: $e = (h, M, F)$

1. h_e is the canonical word for the entity (typically the first mention's head word)

Text

A truck bomb exploded near the embassy.
Three militia planted it, and then they fled.

Entity Representation

entity 1: h = *bomb*, F = {PHYS-OBJ},
M = { (p=explode, d=subject-explode)
(p=plant, d=object-plant) }
entity 2: h = *militia*, F = {PERSON, ORG},
M = { (p=plant, d=subject-plant),
(p=flee, subject-flee) }
entity 3: h = *embassy*, F = {PHYS-OBJ, ORG},
M = { (p=explode, d=prep_near-explode) }

Figure 2: Example text mapped to our entities.

2. M_e is a set of entity mentions $m \in M_e$. Each mention is a pair $m = (p, d)$: the predicate, and the typed dependency from the predicate to the mention (e.g., *push* and *subject-push*).

3. F_e is a set of binary entity features. This paper only uses named entity types as features, but generalizes to other features as well.

A document is thus reduced to its entities, their grammatical contexts, and entity features. Figure 2 continues our example using this formulation. h_e is chosen to be e's longest non-pronoun mention $m \in M_e$. Mentions are labeled with NER and WordNet synsets to create an entity's features $F_e \subseteq \{$*Person, Org, Loc, Event, Time, Object, Other*$\}$. We use the Stanford NLP toolkit to parse, extract typed dependencies, label with NER, and run coreference.

4.3 The Generative Models

Similar to topics in LDA, each document d in our model has a corresponding multinomial over *schema types* θ_d, drawn from a Dirichlet. For each entity in the document, a hidden variable t is drawn according to θ_d. These t variables represent the high level schema types, such as *bombing* or *kidnapping*. The predicates associated with each of the entity's mentions are then drawn from the schema's multinomial over predicates P_t. The variable t also generates a hidden variable s from its distribution over slots, such as *perpetrator* and *victim*. Finally, the entity's canonical head word is generated from β_s, all entity mentions' typed dependencies from δ_s, and named entity types from γ_s.

The most important characteristic of this model is the separation of event words from the lexical properties of specific entity mentions. The schema type variables t only model the distribution of event words (bomb, plant, defuse), but the slot variables s model the syntax (subject-bomb, subject-plant, object-arrest) and entity words (suspect, terrorist, man). This allows the high-level schemas to first select predicates, and then forces predicate arguments to prefer slots that are in the parent schema type.

Formally, a document d receives a labeling Z_d where each entity $e \in E_d$ is labeled $Z_{d,e} = (t, s)$ with a schema type t and a slot s. The joint distribution of a document and labeling is then as follows:

$$
\begin{aligned}
P(d, Z_d) = & \prod_{e \in E_d} P(t|\theta) \times P(s|t) \\
& \times \prod_{e \in E_d} P(h_e|s) \\
& \times \prod_{e \in E_d} \prod_{f \in F_e} P(f|s) \\
& \times \prod_{e \in E_d} \prod_{m \in M_e} P(d_m|s) * P(p_m|t) \qquad (1)
\end{aligned}
$$

The plate diagram for the model is given in Figure 3. The darker circles correspond to the observed entity components in Figure 2. We assume the following generative process for a document d:

Generate θ_d from Dir(α)
for each schema type $t = 1...m$ **do**
  Generate P_t from Dir(η)
  for each slot $s_t = 1...k$ **do**
    Generate β_s from Dir(μ)
    Generate γ_s from Dir(ν)
    Generate δ_s from Dir(φ)
for each entity $e \in E_d$ **do**
  Generate schema type t from Multinomial(θ_d)
  Generate slot s from UniformDist(k)
  Generate head word h from Multinomial(β_s)
  for each mention $m \in M_e$ **do**
    Generate predicate token p from Multinomial(P_t)
    Generate typed dependency d from Multinomial(δ_s)
  for each entity type $i = 1..|F_e|$ **do**
    Generate entity type f from Multinomial(γ_s)

The number of schema types m and the number of slots per schema k are chosen based on training set performance.

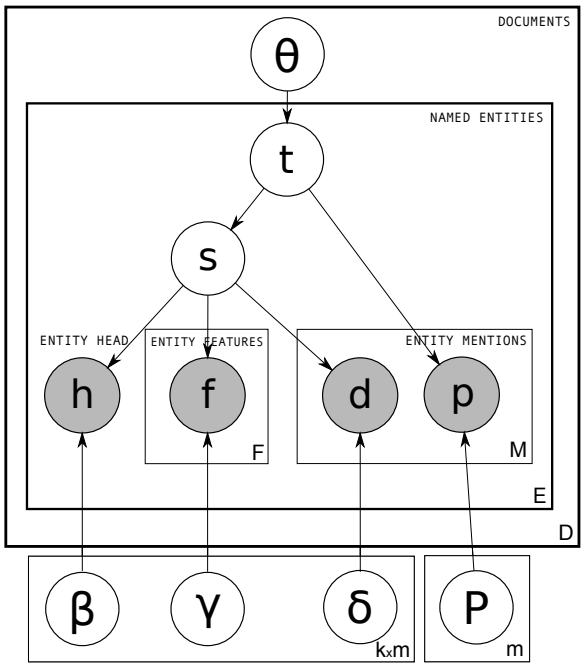

Figure 3: The full plate diagram for the event schema model. Hyper-parameters are omitted for readability.

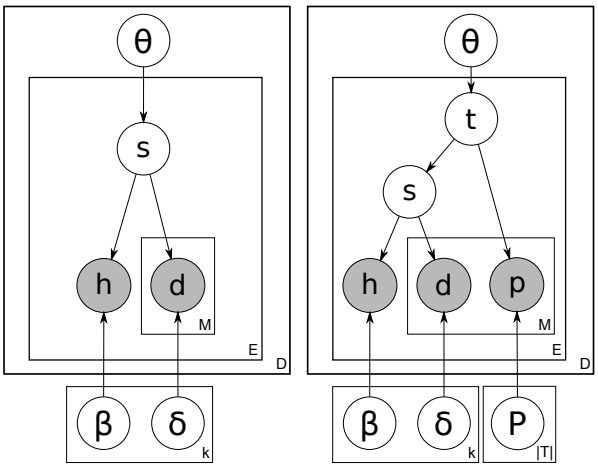

Figure 4: Simplified plate diagrams comparing the flat relation model to the full template model. The observed $f \in F$ variables are not included for clarity.

The Flat Relation Model

We also experiment with a *Flat Relation Model* that removes the hidden t variables, ignoring schema types. Figure 4 visually compares this flat model with the full model. We found that the predicate distribution P_t hurts performance in a flat model. Predicates are more informative at the higher level, but less so for slots where syntax is more important. We thus removed P_t from the model, and everything else remains the same. This flat model now learns a large set of k slots S that aren't connected by a high-level schema variable. Each slot $s \in S$ has a corresponding triple of multinomials (h, M, F) similar to above: (1) a multinomial over the head mentions β_s, (2) a multinomial over the grammatical relations of the entity mentions δ_s, and (3) a multinomial over the entity features γ_s. For each entity in a document, a hidden slot $s \in S$ is first drawn from Θ, and then the observed entity (h, M, F) is drawn according to the multinomials $(\beta_s, \gamma_s, \delta_s)$. We later evaluate this flat model to show the benefit of added schema structure.

4.4 Inference

We use collapsed Gibbs sampling for inference, sampling the latent variables $t_{e,d}$ and $s_{e,d}$ in sequence conditioned on a full setting of all the other variables (Griffiths and Steyvers, 2004). Initial parameter values are set by randomly setting t and s variables from the uniform distribution over schema types and slots, then computing the other parameter values based on these initial settings. The hyperparameters for the dirichlet distributions were chosen from a small grid search (see Experiments).

Beyond standard inference, we added one constraint to the model that favors grammatical distributions δ_s that do not contain conflicts. The subject and direct object of a verb should not both receive high probability mass under the same schema slot δ_s. For instance, the *victim* of a *kidnapping* should not favor both the subject and object of a single verb. Semantic roles should (typically) select one syntactic slot, so this constraint encourages that behavior. During sampling of $s_{e,d}$, we use a penalty factor λ to make conflicting relations less likely. Formally, $P(s_{e,d} = s|\theta, h_e, F_e, M_e) = \lambda$ iff there exists an $m \in M_e$ such that $P(m|\sigma_s) < P(inv(m)|\sigma_s)$ and $P(inv(m)|\sigma_s) > 0.1$, where $inv(m) = object$ if $m = subject$ and vice versa. Otherwise, the probability is computed as normal. We normalize the distributions after penalties are computed.

4.5 Entity Extraction for Template Filling

Inducing event schemas is only one benefit of the model. The learned model can also extract specific instances of the learned schemas without ad-

ditional complexity. To evaluate the effectiveness of the model, we apply the model to perform standard template extraction on MUC-4. Previous MUC-4 induction required an extraction algorithm separate from induction because induction created hard clusters (Chambers and Jurafsky, 2011). Cluster scores don't have a natural interpretation, so extraction required several parameters/thresholds to tune. Our model instead simply relies on model inference.

We run inference as described above and each entity receives a template label $t_{e,d}$ and a template slot label $s_{e,d}$. These labels are the extractions, and it requires no other parameters. The model thus requires far less machinery than a pipeline, and the experiments below further show that this simpler model outperforms the pipeline.

Beyond parameters, the question of "irrelevant" documents is a concern in MUC-4. Approximately half the corpus are documents that are not labeled with a template, so past algorithms required extra processing stages to filter out these irrelevant documents. Patwardhan and Riloff (2009) and Chambers and Jurafsky (2011) make initial decisions as to whether they should extract or not from a document. Huang and Riloff (2011) use a genre detector for this problem. Even the generative HMM-based model of Cheung et al. (Cheung et al., 2013) requires an extra filtering parameter. Our formal model is unique in not requiring additional effort. Ours is the only approach that doesn't require document filtering.

5 Evaluation Setup

Evaluating on MUC-4 has a diverse history that complicates comparison. The following balances comparison against previous work and enables future comparison to our results.

5.1 Template Schema Slots

Most systems do not evaluate performance on all MUC-4 template slots. They instead focus on four main slots, ignoring the parameterized slots that involve deeper reasoning (such as 'stage of execution' and 'effect of incident'). The four slots and example entity fillers are shown here:

Perpetrator:	Shining Path members
Victim:	Sergio Horna
Target:	public facilities
Instrument:	explosives

We also focus only on these four slots. We merged MUC's two perpetrator slots (individuals and orgs) into one gold Perpetrator. Previous work has both split the two and merged the two. We merge them because the distinction between an individual and an organization is often subtle and not practically important to analysts. This is also consistent with the most recent event schema induction in Chambers and Jurafsky (2011) and Cheung et al. (2013).

One peculiarity in MUC-4 is that some templates are labeled as optional (i.e., all its slots are optional), and some required templates contain optional slots (i.e., a subset of slots are optional). We ignore both optional templates and specific optional slots *when computing recall*, as in previous work (Patwardhan and Riloff, 2007; Patwardhan and Riloff, 2009; Chambers and Jurafsky, 2011).

Comparison between the extracted strings and the gold template strings uses head word scoring. We do not use gold parses for the text, so head words are defined simply as the rightmost word in the noun phrase. The exception is when the extracted phrase is of the form "A of B", then the rightmost word in "A" is used as the head. This is again consistent with previous work[1]. The standard evaluation metrics are precision, recall, and F1 score.

5.2 Mapping Learned Slots

Induced schemas need to map to gold schemas before evaluation. Which learned slots correspond to MUC-4 slots? There are two methods of mapping. The first ignores the schema type variables t, and simply finds the best performing s variable for each gold template slot[2]. We call this the *slot-only mapping* evaluation. The second approach is to map each template variable t to the best gold template type g, and limit the slot mapping so that only the slots under t can map to slots under g. We call this the *template mapping* evaluation. The slot-only mapping can result in higher scores since it is not constrained to preserve schema structure in the mapping.

Chambers and Jurafsky (2011) used template mapping in their evaluation. Cheung et al. (2013) used slot-only mapping. We run both evaluations in this paper and separately compare both.

[1] Personal communications with Patwardhan and Riloff

[2] bombing-victim is a template slot distinct from kidnap-victim. Both need to be mapped.

6 Experiments

We use the Stanford CoreNLP toolkit for text processing and parsing. We developed the models on the 1300 document MUC-4 training set. We then learned once on the entire 1700 training/dev/test set, and report extraction numbers from the inferred labels on the 200 document test set. Each experiment was repeated 10 times. Reported numbers are averaged across these runs.

There are two structure variables for the model: the number of schema types and the number of slots under each type. We searched for the optimal values on the training set before evaluating on test. The hyperparameters for all evaluations were set to $\alpha = \eta = \mu = \nu = 1$, $\varphi = .1$ based on a grid search.

6.1 Template Schema Induction

The first evaluation compares the learned schemas to the gold schemas in MUC-4.

Since most previous work assumes this knowledge ahead of time, we align our schemas with the main MUC-4 template types to measure quality. We inspected the learned event schemas that mapped to MUC-4 schemas based on the *template mapping* extraction evaluation.

Figure 5 shows some of the learned distributions for two mapped schemas: kidnappings and bombings. The predicate distribution for each event schema is shown, as well as the top 5 head words and grammatical relations for each slot. The words and events that were jointly learned in these examples appear quite accurate. The bombing and kidnap schemas learned all of the equivalent MUC-4 gold slots. Interestingly, our model also learned Locations and Times as important entities that appear in the text. These entities are not traditionally included in the MUC-4 extraction task.

Figure 6 lists the MUC-4 slots that we did and did not learn for the four most prevelant types. We report 71% recall, with almost all errors due to the model's failure to learn about arsons. Arson templates only occur in 40 articles, much less than the 200 bombing and over 400 attack. We show below that overall extraction performs well despite this. The learned distributions for Attack end up extracting Arson perpetrators and Arson victims in the actual extraction evaluation.

	Bomb	Kidnap	Attack	Arson
Perpetrator	✓	✓	✓	x
Victim	✓	✓	✓	✓
Target	✓	-	✓	x
Instrument	✓	-	x	x
Location	✓	✓	✓	✓
Date/Time	✓	✓	✓	x

Figure 6: The MUC-4 gold slots that were learned. The bottom two are not in the traditional evaluation, but were learned by our model nonetheless.

Evaluation: Template Mapping

	Prec	Recall	F1
C & J 2011	**.48**	.25	.33
Formal Template Model	.42	**.27**	.33

Table 1: MUC-4 extraction with template mapping. A learned schema first maps to a gold MUC template. Learned slots can then only map to slots in that template.

6.2 Extraction Experiments

We now present the full extraction experiment that is traditionally used for evaluating MUC-4 performance. Although our learned schemas closely match gold schemas, extraction depends on how well the model can extract from diverse lexical contexts. We ran inference on the full training and test sets, and used the inferred labels as schema labels. These labels were mapped and evaluated against the gold MUC-4 labels as discussed in Section 5.

Performance is compared to two state-of-the-art induction systems. Since these previous two models used different methods to map their learned schemas, we compare separately. Table 1 shows the *template mapping* evaluation with Chambers and Jurafsky (C&J). Table 2 shows the *slot-only mapping* evaluation with Cheung et al.

Our model achieves an F1 score comparable to C&J, and 20% higher than Cheung et al. Part of the greater increase over Cheung et al. is the mapping difference. For each MUC-4 type, such as *bombing*, any four learned slots can map to the four MUC-4 bombing slots. There is no constraint that the learned slots must come from the same schema type. The more strict *template mapping* (Table 1) ensures that entire schema types are mapped together, and it reduces our performance from .41 to .33.

Kidnapping Entities

Victim (Person 88%)		**Perpetrator** (Person 62%, Org 30%)		**Date** (TimeDate 89%)	
businessman	object-kidnap	guerrilla	subject-kidnap	TIME	tmod-kidnap
citizen	object-release	ELN	subject-hold	February	prep_on-kidnap
Soares	prep_of-kidnapping	group	subject-attack	hours	tmod-release
Kent	possessive-release	extraditables	subject-demand	morning	prep_on-release
hostage	object-found	man	subject-announce	night	tmod-take

Bombing Entities

Victim (Person 86%, Location 8%)		**Physical Target** (Object 65%, Event 42%)		**Instrument** (Event 56%, Object 39%)	
person	object-kill	building	object-destroy	bomb	subject-explode
guerrilla	object-wound	office	object-damage	explosion	subject-occur
soldier	subject-die	explosive	object-use	attack	object-cause
man	subject-blow_up	station	and-office	charge	object-place
civilian	subject-try	vehicle	prep_of-number	device	subject-destroy

Figure 5: Select distributions for two learned events. Left columns are head word distributions β, right columns are syntactic relation distributions δ, and entity types in parentheses are the learned γ. Most probable words are shown.

Evaluation: Slot-Only Mapping

	Prec	Recall	F1
Cheung et al. 2013	.32	.37	.34
Flat Relation Model	.26	**.45**	.33
Formal Template Model	**.41**	.41	**.41**

Table 2: MUC-4 extraction with slot-only mapping. Any learned slot is allowed to map to any gold slot.

Entity Role Performance

	Prec	Recall	F1
Perpetrator	.40	.20	.26
Victim	.42	.31	.34
Target	.38	.28	.31
Instrument	.57	.39	.45

Table 3: Results for each MUC-4 template slot using the template-mapping evaluation.

The macro-level F1 scores can be broken down into individual slot performance. Table 3 shows these results ranging from .26 to .45. The Instrument role proves easiest to learn, consistent with C&J.

A large portion of MUC-4 includes irrelevant documents. Cheung et al. (2013) evaluated their model without irrelevant documents in the test set that to see how performance is affected. We compare against their numbers in Table 4. Results are closer now with ours outperforming .46 to .43 F1. This suggests that the HMM-based approach stumbles more on spurious documents, but performs better on relevant ones.

Gold Document Evaluation

	Prec	Recall	F1
Cheung et al. 2013	.41	**.44**	.43
Formal Template Model	**.49**	.43	**.46**

Table 4: Full MUC-4 extraction with *gold document classification*. These results ignore false positives extracted from "irrelevant" documents in the test set.

6.3 Model Ablation

Table 2 shows that the flat relation model (no latent type variables t) is inferior to the full schema model. F1 drops 20% without the explicit modeling of both schema types t and their entity slots s. The entity features F_e are less important. Experiments without them show a slight drop in performance (2 F1 points), small enough that they could be removed for efficiency. However, it is extremely useful to learn slots with NER labels like Person or Location.

Finally, we experimented without the subject/object constraint (Section 4.4). Performance drops 5-10% depending on the number of schemas learned. Anecdotally, it merges too many schema slots that should be separate. We recommend using this constraint as it has little impact on CPU time.

6.4 Extension: Reduce Training Size

One of the main benefits of this generative model appears to be the reduction in training data. The pipelined approach in C&J required an information retrieval stage to bring in hundreds of other docu-

ments from an external corpus. This paper's generative model doesn't require such a stage.

We thus attempted to induce and extract event schemas from just the 200 test set documents, with no training or development data. We repeated this experiment 30 times and averaged the results, setting the number of templates $t = 20$ and slots $s = 10$ as in the main experiment. The resulting F1 score for the template-mapping evaluation fell to 0.27 from the full data experiment of 0.33 F1. Adding more training documents in another experiment did not significantly increase performance over 0.27 until all training and development documents were included. This could be explained by the development set being more similar to the test set than training. We did not investigate further to prevent over-experimentation on test.

7 Discussion

Our model is one of the first generative formulations of schema induction. It produces state-of-the-art performance on a traditional extraction task, and performs with less training data as well as a more complex pipelined approach. Further, our unique entity-driven approach outperforms an HMM-based model developed in parallel to this work.

Our entity-driven proposal is strongly influenced by the ideas in the pipeline model of Chambers and Jurafsky (2011). Coreference chains have been used in a variety of learning tasks, such as narrative learning and summarization. Here we are the first to show how it can be used for schema induction in a probabilistic model, connecting predicates across a document in a way that is otherwise difficult to represent. The models perform similarly, but ours also includes significant benefits like a reduction in complexity, reproducibility, and a large reduction in training data requirements.

This paper also implies that learning and extraction need not be independent algorithms. Our model's inference procedure to learn schemas is the same one that labels text for extraction. C&J required 3-4 separate pipelined steps. Cheung et al. (2013) required specific cutoffs for document classification before extraction. Not only does our model perform well, but it does so without these steps.

Highlighted here are key differences between this proposal and the HMM-based model of Cheung et al. (2013). One of the HMM strengths is the inclusion of sequence-based knowledge. Each slot label is influenced by the previous label in the text, encouraging syntactic arguments of a predicate to choose the same schema. This knowledge is only loosely present in our document distribution θ. Cheung et al. also include a hidden event variable between the template and slot variables. Our model collapses this event variable and makes fewer dependency assumptions. This difference requires further investigation as it is unclear if it provides valuable information, or too much complexity.

We also note a warning for future work on proper evaluation methodology. This task is particularly difficult to compare to other models due to its combination of both induction and then extraction. There are many ways to map induced schemas to gold answers, and this paper illustrates how extraction performance is significantly affected by the choice. We suggest the template-mapping evaluation to preserve learned structure.

Finally, these induced results are far behind supervised learning (Huang and Riloff, 2011). There is ample room for improvement and future research in event schema induction.

Acknowledgments

This work was partially supported by a grant from the Office of Naval Research. It was also supported, in part, by the Johns Hopkins Human Language Technology Center of Excellence. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author. Thanks to Eric Wang for his insights into Bayesian modeling, Brendan O'Connor for his efforts on normalizing MUC-4 evaluation details, Frank Ferraro and Benjamin Van Durme for helpful conversations, and to the reviewers for insightful feedback.

References

Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007a. Learning relations from the web. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007b. Open information extraction from the web. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*.

Razvan Bunescu and Raymond Mooney. 2004. Collective information extraction with relational markov networks. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 438–445.

Andrew Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R. Hruschka Jr., and T.M. Mitchell. 2010a. Toward an architecture for never-ending language learning. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*.

Andrew Carlson, J. Betteridge, R.C. Wang, E.R. Hruschka Jr., and T.M. Mitchell. 2010b. Coupled semi-supervised learning for information extraction. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*.

Nathanael Chambers and Dan Jurafsky. 2011. Template-based information extraction without the templates. In *Proceedings of the Association for Computational Linguistics*.

Harr Chen, Edward Benson, Tahira Naseem, and Regina Barzilay. 2011. In-domain relation discovery with meta-constraints via posterior regularization. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. 2013. Probabilistic frame induction. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Hai Leong Chieu, Hwee Tou Ng, and Yoong Keok Lee. 2003. Closing the gap: Learning-based information extraction rivaling knowledge-engineering methods. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Nancy Chinchor, David Lewis, and Lynette Hirschman. 1993. Evaluating message understanding systems: an analysis of the third message understanding conference. *Computational Linguistics*, 19:3:409–449.

Benjamin Van Durme and Marius Pasca. 2008. Finding cars, goddesses and enzymes: Parametrizable acquisition of labeled instances for open-domain information extraction. In *Proceedings of the 23rd Annual Conference on Artificial Intelligence (AAAI-2008)*, pages 1243–1248.

Elena Filatova, Vasileios Hatzivassiloglou, and Kathleen McKeown. 2006. Automatic creation of domain templates. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Dayne Freitag. 1998. Toward general-purpose learning for information extraction. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 404–408.

T. L. Griffiths and M. Steyvers. 2004. Finding scientific topics. In *Proceedings of the National Academy of Sciences of the United States of America*, pages 5228–5235.

Aria Haghighi and Dan Klein. 2010. An entity-level approach to information extraction. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Ruihong Huang and Ellen Riloff. 2010. Inducing domain-specific semantic class taggers from (almost) nothing. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Ruihong Huang and Ellen Riloff. 2011. Peeling back the layers: Detecting event role fillers in secondary contexts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Mstislav Maslennikov and Tat-Seng Chua. 2007. Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Brendan O'Connor. 2012. Learning frames from text with an unsupervised latent variable model. Technical report, Carnegie Mellon University.

Siddharth Patwardhan and Ellen Riloff. 2007. Effective ie with semantic affinity patterns and relevant regions. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP)*.

Siddharth Patwardhan and Ellen Riloff. 2009. A unified model of phrasal and sentential evidence for information extraction. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP)*.

Lisa Rau, George Krupka, Paul Jacobs, Ira Sider, and Lois Childs. 1992. Ge nltoolset: Muc-4 test results and analysis. In *Proceedings of the Message Understanding Conference (MUC-4)*, pages 94–99.

Ellen Riloff and Mark Schmelzenbach. 1998. An empirical approach to conceptual case frame acquisition. In *Proceedings of the Sixth Workshop on Very Large Corpora*.

Ellen Riloff, Janyce Wiebe, and William Phillips. 2005. Exploiting subjectivity classification to improve information extraction. In *Proceedings of AAAI-05*.

Satoshi Sekine. 2006. On-demand information extraction. In *Proceedings of the Joint Conference of the*

International Committee on Computational Linguistics and the Association for Computational Linguistics, pages 731–738.

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive ie using unrestricted relation discovery. In *Proceedings of NAACL*.

Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. 2003. An improved extraction pattern representation model for automatic ie pattern acquisition. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 224–231.

Beth M. Sundheim. 1991. Third message understanding evaluation and conference (muc-3): Phase 1 status report. In *Proceedings of the Message Understanding Conference*.

Mihai Surdeanu, Jordi Turmo, and Alicia Ageno. 2006. A hybrid approach for the acquisition of information extraction patterns. In *Proceedings of the EACL Workshop on Adaptive Text Extraction and Mining*.

Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. 2000. Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING)*, pages 940–946.