In-domain relation discovery with meta-constraints via posterior regularization

In-domain Relation Discovery with Meta-constraints via Posterior Regularization

Harr Chen, Edward Benson, Tahira Naseem, and **Regina Barzilay**
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
{harr, eob, tahira, regina} @csail.mit.edu

Abstract

We present a novel approach to discovering relations and their instantiations from a collection of documents in a single domain. Our approach learns relation types by exploiting *meta-constraints* that characterize the general qualities of a good relation in any domain. These constraints state that instances of a single relation should exhibit regularities at multiple levels of linguistic structure, including lexicography, syntax, and document-level context. We capture these regularities via the structure of our probabilistic model as well as a set of declaratively-specified constraints enforced during posterior inference. Across two domains our approach successfully recovers hidden relation structure, comparable to or outperforming previous state-of-the-art approaches. Furthermore, we find that a small set of constraints is applicable across the domains, and that using domain-specific constraints can further improve performance. [1]

1 Introduction

In this paper, we introduce a novel approach for the unsupervised learning of relations and their instantiations from a set of in-domain documents. Given a collection of news articles about earthquakes, for example, our method discovers relations such as the earthquake's location and resulting damage, and extracts phrases representing the relations' instantiations. Clusters of similar in-domain documents are increasingly available in forms such as Wikipedia article categories, financial reports, and biographies.

A strong earthquake rocked the Philippine island of Mindoro early Tuesday, [destroying]$_{ind}$ [some homes]$_{arg}$ ...
A strong earthquake hit the China-Burma border early Wednesday ... The official Xinhua News Agency said [some houses]$_{arg}$ were [damaged]$_{ind}$ ...
A strong earthquake with a preliminary magnitude of 6.6 shook northwestern Greece on Saturday, ... [destroying]$_{ind}$ [hundreds of old houses]$_{arg}$ ...

Figure 1: Excerpts from newswire articles about earthquakes. The indicator and argument words for the *damage* relation are highlighted.

In contrast to previous work, our approach learns from *domain-independent meta-constraints* on relation expression, rather than supervision specific to particular relations and their instances. In particular, we leverage the linguistic intuition that documents in a single domain exhibit regularities in how they express their relations. These regularities occur both in the relations' lexical and syntactic realizations as well as at the level of document structure. For instance, consider the *damage* relation excerpted from earthquake articles in Figure 1. Lexically, we observe similar words in the instances and their contexts, such as "destroying" and "houses." Syntactically, in two instances the relation instantiation is the dependency child of the word "destroying." On the discourse level, these instances appear toward the beginning of their respective documents. In general, valid relations in many domains are characterized by these coherence properties.

We capture these regularities using a Bayesian model where the underlying relations are repre-

[1] The source code for this work is available at: http://groups.csail.mit.edu/rbg/code/relation_extraction/

sented as latent variables. The model takes as input a constituent-parsed corpus and explains how the constituents arise from the latent variables. Each relation instantiation is encoded by the variables as a relation-evoking *indicator* word (*e.g.*, "destroying") and corresponding *argument* constituent (*e.g.*, "some homes").[2] Our approach capitalizes on relation regularity in two ways. First, the model's generative process encourages coherence in the local features and placement of relation instances. Second, we apply posterior regularization (Graça et al., 2007) during inference to enforce higher-level declarative constraints, such as requiring indicators and arguments to be syntactically linked.

We evaluate our approach on two domains previously studied for high-level document structure analysis, news articles about earthquakes and financial markets. Our results demonstrate that we can successfully identify domain-relevant relations. We also study the importance and effectiveness of the declaratively-specified constraints. In particular, we find that a small set of declarative constraints are effective across domains, while additional domain-specific constraints yield further benefits.

2 Related Work

Extraction with Reduced Supervision Recent research in information extraction has taken large steps toward reducing the need for labeled data. Examples include using bootstrapping to amplify small seed sets of example outputs (Agichtein and Gravano, 2000; Yangarber et al., 2000; Bunescu and Mooney, 2007; Zhu et al., 2009), leveraging existing databases that overlap with the text (Mintz et al., 2009; Yao et al., 2010), and learning general domain-independent knowledge bases by exploiting redundancies in large web and news corpora (Hasegawa et al., 2004; Shinyama and Sekine, 2006; Banko et al., 2007; Yates and Etzioni, 2009).

Our approach is distinct in both the supervision and data we operate over. First, in contrast to bootstrapping and database matching approaches, we learn from meta-qualities, such as low variability in syntactic patterns, that characterize a good relation. We hypothesize that these properties hold across relations in different domains. Second, in contrast to work that builds general relation databases from heterogeneous corpora, our focus is on learning the relations salient in a single domain. Our setup is more germane to specialized domains expressing information not broadly available on the web.

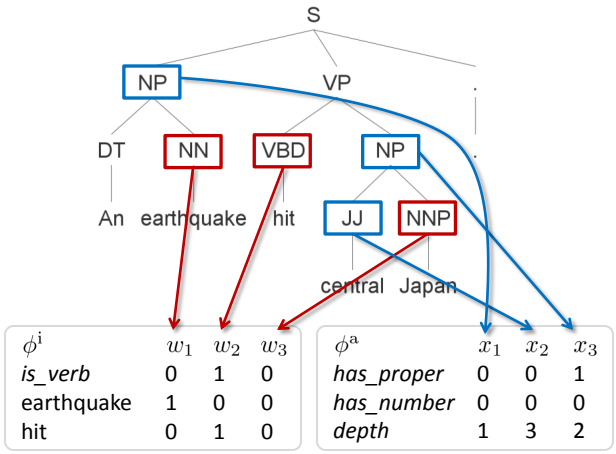
Earlier work in unsupervised information extraction has also leveraged meta-knowledge independent of specific relation types, such as declaratively-specified syntactic patterns (Riloff, 1996), frequent dependency subtree patterns (Sudo et al., 2003), and automatic clusterings of syntactic patterns (Lin and Pantel, 2001; Zhang et al., 2005) and contexts (Chen et al., 2005; Rosenfeld and Feldman, 2007). Our approach incorporates a broader range of constraints and balances constraints with underlying patterns learned from the data, thereby requiring more sophisticated machinery for modeling and inference.

Extraction with Constraints Previous work has recognized the appeal of applying declarative constraints to extraction. In a supervised setting, Roth and Yih (2004) induce relations by using linear programming to impose global declarative constraints on the output from a set of classifiers trained on local features. Chang et al. (2007) propose an objective function for semi-supervised extraction that balances likelihood of labeled instances and constraint violation on unlabeled instances. Recent work has also explored how certain kinds of supervision can be formulated as constraints on model posteriors. Such constraints are not declarative, but instead based on annotations of words' majority relation labels (Mann and McCallum, 2008) and pre-existing databases with the desired output schema (Bellare and McCallum, 2009). In contrast to previous work, our approach explores a different class of constraints that does not rely on supervision that is specific to particular relation types and their instances.

3 Model

Our work performs in-domain relation discovery by leveraging regularities in relation expression at the lexical, syntactic, and discourse levels. These regularities are captured via two components: a probabilistic model that explains how documents are generated from latent relation variables and a technique

[2] We do not use the word "argument" in the syntactic sense—a relation's argument may or may not be the syntactic dependency argument of its indicator.



ϕ^{i}	w_1	w_2	w_3
is_verb	0	1	0
earthquake	1	0	0
hit	0	1	0

ϕ^{a}	x_1	x_2	x_3
has_proper	0	0	1
has_number	0	0	0
depth	1	3	2

Figure 2: Words w and constituents x of syntactic parses are represented with indicator features ϕ^{i} and argument features ϕ^{a} respectively. A single relation instantiation is a pair of indicator w and argument x; we filter w to be nouns and verbs and x to be noun phrases and adjectives.

for biasing inference to adhere to declaratively-specified constraints on relation expression. This section describes the generative process, while Sections 4 and 5 discuss declarative constraints.

3.1 Problem Formulation

Our input is a corpus of constituent-parsed documents and a number K of relation types. The output is K clusters of semantically related relation instantiations. We represent these instantiations as a pair of *indicator* word and *argument* sequence from the same sentence. The indicator's role is to anchor a relation and identify its type. We only allow nouns or verbs to be indicators. For instance, in the earthquake domain a likely indicator for *damage* would be "destroyed." The argument is the actual relation value, *e.g.*, "some homes," and corresponds to a noun phrase or adjective.[3]

Along with the document parse trees, we utilize a set of features $\phi^{\text{i}}(w)$ and $\phi^{\text{a}}(x)$ describing each potential indicator word w and argument constituent x, respectively. An example feature representation is shown in Figure 2. These features can encode words, part-of-speech tags, context, and so on. Indicator and argument feature definitions need not be the same (*e.g.*, *has_number* is important for arguments but irrelevant for indicators).[4]

3.2 Generative Process

Our model associates each relation type k with a set of *feature distributions* θ_k and a *location distribution* λ_k. Each instantiation's indicator and argument, and its position within a document, are drawn from these distributions. By sharing distributions within each relation, the model places high probability mass on clusters of instantiations that are coherent in features and position. Furthermore, we allow at most one instantiation per document and relation, so as to target relations that are relevant to the entire document.

There are three steps to the generative process. First, we draw feature and location distributions for each relation. Second, an instantiation is selected for every pair of document d and relation k. Third, the indicator features of each word and argument features of each constituent are generated based on the relation parameters and instantiations. Figure 3 presents a reference for the generative process.

Generating Relation Parameters Each relation k is associated with four feature distribution parameter vectors: θ^{i}_k for indicator words, θ^{bi}_k for non-indicator words, θ^{a}_k for argument constituents, and θ^{ba}_k for non-argument constituents. Each of these is a set of multinomial parameters per feature drawn from a symmetric Dirichlet prior. A likely indicator word should have features that are highly probable according to θ^{i}_k, and likewise for arguments and θ^{a}_k. Parameters θ^{bi}_k and θ^{ba}_k represent *background* distributions for non-relation words and constituents, similar in spirit to other uses of background distributions that filter out irrelevant words (Chemudugunta et al., 2006).[5] By drawing each instance from these distributions, we encourage the relation to be coherent in local lexical and syntactic properties.

Each relation type k is also associated with a parameter vector λ_k over document segments drawn from a symmetric Dirichlet prior. Documents are divided into L equal-length segments; λ_k states how likely relation k is for each segment, with one null outcome for the relation not occurring in the document. Because λ_k is shared within a relation, its

[3] In this paper we focus on unary relations; binary relations can be modeled with extensions of the hidden variables and constraints.

[4] We consider only categorical features here, though the extension to continuous or ordinal features is straightforward.

[5] We use separate background distributions for each relation to make inference more tractable.

For each relation type k:

- For each indicator feature ϕ^{i} draw feature distributions $\theta^{\text{i}}_{k,\phi^{\text{i}}}, \theta^{\text{bi}}_{k,\phi^{\text{i}}} \sim \text{Dir}(\theta_0)$
- For each argument feature ϕ^{a} draw feature distributions $\theta^{\text{a}}_{k,\phi^{\text{a}}}, \theta^{\text{ba}}_{k,\phi^{\text{a}}} \sim \text{Dir}(\theta_0)$
- Draw location distribution $\lambda_k \sim \text{Dir}(\lambda_0)$

For each relation type k and document d:

- Select document segment $s_{d,k} \sim \text{Mult}(\lambda_k)$
- Select sentence $z_{d,k}$ uniformly from segment $s_{d,k}$, and indicator $i_{d,k}$ and argument $a_{d,k}$ uniformly from sentence $z_{d,k}$

For each word w in every document d:

- Draw each indicator feature $\phi^{\text{i}}(w) \sim \text{Mult}\left(\frac{1}{Z}\prod_{k=1}^{K}\theta_{k,\phi^{\text{i}}}\right)$, where $\theta_{k,\phi^{\text{i}}}$ is $\theta^{\text{i}}_{k,\phi^{\text{i}}}$ if $i_{d,k} = w$ and $\theta^{\text{bi}}_{k,\phi^{\text{i}}}$ otherwise

For each constituent x in every document d:

- Draw each argument feature $\phi^{\text{a}}(x) \sim \text{Mult}\left(\frac{1}{Z}\prod_{k=1}^{K}\theta_{k,\phi^{\text{a}}}\right)$, where $\theta_{k,\phi^{\text{a}}}$ is $\theta^{\text{a}}_{k,\phi^{\text{a}}}$ if $a_{d,k} = x$ and $\theta^{\text{ba}}_{k,\phi^{\text{a}}}$ otherwise

Figure 3: The generative process for model parameters and features. In the above Dir and Mult refer respectively to the Dirichlet distribution and multinomial distribution. Fixed hyperparameters are subscripted with zero.

instances will tend to occur in the same relative positions across documents. The model can learn, for example, that a particular relation typically occurs in the first quarter of a document (if $L = 4$).

Generating Relation Instantiations For every relation type k and document d, we first choose which portion of the document (if any) contains the instantiation by drawing a document segment $s_{d,k}$ from λ_k. Our model only draws one instantiation per pair of k and d, so each discovered instantiation within a document is a separate relation. We then choose the specific sentence $z_{d,k}$ uniformly from within the segment, and the indicator word $i_{d,k}$ and argument constituent $a_{d,k}$ uniformly from within that sentence.

Generating Text Finally, we draw the feature values. We make a Naïve Bayes assumption between features, drawing each independently conditioned on relation structure. For a word w, we want all relations to be able to influence its generation. Toward this end, we compute the element-wise product of feature parameters across relations $k = 1, \ldots, K$, using indicator parameters θ^{i}_k if relation k selected w as an indicator word (if $i_{d,k} = w$) and background parameters θ^{bi}_k otherwise. The result is then normalized to form a valid multinomial that produces word w's features. Constituents are drawn similarly from every relations' argument distributions.

4 Inference with Constraints

The model presented above leverages relation regularities in local features and document placement. However, it is unable to specify global syntactic preferences about relation expression, such as indicators and arguments being in the same clause. Another issue with this model is that different relations could overlap in their indicators and arguments.[6]

To overcome these obstacles, we apply declarative constraints by imposing inequality constraints on expectations of the posterior during inference using *posterior regularization* (Graça et al., 2007). In this section we present the technical details of the approach; Section 5 explains the specific linguistically-motivated constraints we consider.

4.1 Inference with Posterior Regularization

We first review how posterior regularization impacts the variational inference procedure in general. Let θ, z, and x denote the parameters, hidden structure, and observations of an arbitrary model. We are interested in estimating the posterior distribution $p(\theta, z \mid x)$ by finding a distribution $q(\theta, z) \in \mathcal{Q}$ that is minimal in KL-divergence to the true posterior:

$$\begin{aligned}&\text{KL}(q(\theta, z) \parallel p(\theta, z \mid x)) \\ &= \int q(\theta, z) \log \frac{q(\theta, z)}{p(\theta, z, x)} d\theta dz + \log p(x). \quad (1)\end{aligned}$$

For tractability, variational inference typically makes a mean-field assumption that restricts the set $\mathcal{Q}$ to distributions where θ and z are independent, *i.e.*, $q(\theta, z) = q(\theta)q(z)$. We then optimize equation 1 by coordinate-wise descent on $q(\theta)$ and $q(z)$.

To incorporate constraints into inference, we further restrict $\mathcal{Q}$ to distributions that satisfy a given

[6] In fact, a true *maximum a posteriori* estimate of the model parameters would find the same most salient relation over and over again for every k, rather than finding K different relations.

set of inequality constraints, each of the form $\mathbb{E}_q[f(z)] \leq b$. Here, $f(z)$ is a deterministic function of z and b is a user-specified threshold. Inequalities in the opposite direction simply require negating $f(z)$ and b. For example, we could apply a syntactic constraint of the form $\mathbb{E}_q[f(z)] \geq b$, where $f(z)$ counts the number of indicator/argument pairs that are syntactically connected in a pre-specified manner (*e.g.*, the indicator and argument modify the same verb), and b is a fixed threshold.

Given a set $\mathcal{C}$ of constraints with functions $f_c(z)$ and thresholds b_c, the updates for $q(\theta)$ and $q(z)$ from equation 1 are as follows:

$$q(\theta) = \underset{q(\theta)}{\operatorname{argmin}} \operatorname{KL}\big(q(\theta) \parallel q'(\theta)\big), \tag{2}$$

where $q'(\theta) \propto \exp \mathbb{E}_{q(z)}[\log p(\theta, z, x)]$, and

$$q(z) = \underset{q(z)}{\operatorname{argmin}} \operatorname{KL}\big(q(z) \parallel q'(z)\big) \quad s.t. \quad \mathbb{E}_{q(z)}[f_c(z)] \leq b_c, \ \ \forall c \in \mathcal{C}, \tag{3}$$

where $q'(z) \propto \exp \mathbb{E}_{q(\theta)}[\log p(\theta, z, x)]$. Equation 2 is not affected by the posterior constraints and is updated by setting $q(\theta)$ to $q'(\theta)$. We solve equation 3 in its dual form (Graça et al., 2007):

$$\underset{\kappa}{\operatorname{argmin}} \sum_{c \in \mathcal{C}} \kappa_c b_c + \log \sum_z q'(z) e^{-\sum_{c \in \mathcal{C}} \kappa_c f_c(z)} \quad s.t. \quad \kappa_c \geq 0, \ \ \forall c \in \mathcal{C}. \tag{4}$$

With the box constraints of equation 4, a numerical optimization procedure such as L-BFGS-B (Byrd et al., 1995) can be used to find optimal dual parameters κ^*. The original $q(z)$ is then updated to $q'(z) \exp\big(-\sum_{c \in \mathcal{C}} \kappa_c^* f_c(z)\big)$ and renormalized.

4.2 Updates for our Model

Our model uses this mean-field factorization:

$$q(\theta, \lambda, z, a, i) = \prod_{k=1}^{K} q(\lambda_k; \hat{\lambda}_k) q(\theta_k^{\text{i}}; \hat{\theta}_k^{\text{i}}) q(\theta_k^{\text{bi}}; \hat{\theta}_k^{\text{bi}}) q(\theta_k^{\text{a}}; \hat{\theta}_k^{\text{a}}) q(\theta_k^{\text{ba}}; \hat{\theta}_k^{\text{ba}}) \times \prod_d q(z_{d,k}, a_{d,k}, i_{d,k}; \hat{c}_{d,k}) \tag{5}$$

In the above, $\hat{\lambda}$ and $\hat{\theta}$ are Dirichlet distribution parameters, and $\hat{c}$ are multinomial parameters. Note that we do not factorize the distribution of z, i, and a for a single document and relation, instead representing their joint distribution with a single set of variational parameters $\hat{c}$. This is tractable because a single relation occurs only once per document, reducing the joint search space of z, i, and a. The factors in equation 5 are updated one at a time while holding the other factors fixed.

Updating $\hat{\theta}$ Due to the Naïve Bayes assumption between features, each feature's $q(\theta)$ distributions can be updated separately. However, the product between feature parameters of different relations introduces a nonconjugacy in the model, precluding a closed form update. Instead we numerically optimize equation 1 with respect to each $\hat{\theta}$, similarly to previous work (Boyd-Graber and Blei, 2008). For instance, $\hat{\theta}^{\text{i}}_{k,\phi}$ of relation k and feature ϕ is updated by finding the gradient of equation 1 with respect to $\hat{\theta}^{\text{i}}_{k,\phi}$ and applying L-BFGS. Parameters $\hat{\theta}^{\text{bi}}$, $\hat{\theta}^{\text{a}}$, and $\hat{\theta}^{\text{ba}}$ are updated analogously.

Updating $\hat{\lambda}$ This update follows the standard closed form for Dirichlet parameters:

$$\hat{\lambda}_{k,\ell} = \lambda_0 + \mathbb{E}_{q(z,a,i)}[C_\ell(z, a, i)], \tag{6}$$

where C_ℓ counts the number of times z falls into segment ℓ of a document.

Updating $\hat{c}$ Parameters $\hat{c}$ are updated by first computing an unconstrained update $q'(z, a, i; \hat{c}')$:

$$\hat{c}'_{d,k,(z,a,i)} \propto \exp\Bigg(\mathbb{E}_{q(\lambda_k)}[\log p(z, a, i \mid \lambda_k)] + \mathbb{E}_{q(\theta_k^{\text{i}})}[\log p(i \mid \theta_k^{\text{i}})] + \sum_{w \neq i} \mathbb{E}_{q(\theta_k^{\text{bi}})}[\log p(w \mid \theta_k^{\text{bi}})] + \mathbb{E}_{q(\theta_k^{\text{a}})}[\log p(a \mid \theta_k^{\text{a}})] + \sum_{x \neq a} \mathbb{E}_{q(\theta_k^{\text{ba}})}[\log p(x \mid \theta_k^{\text{ba}})] \Bigg)$$

We then perform the minimization on the dual in equation 4 under the provided constraints to derive a final update to the constrained $\hat{c}$.

Simplifying Approximation The update for $\hat{\theta}$ requires numerical optimization due to the nonconjugacy introduced by the point-wise product in feature generation. If instead we have every relation

	Quantity	$f(z, a, i)$	$\leq$ or $\geq$	b
Syntax	$\forall k$	Counts i, a of relation k that match a pattern (see text)	$\geq$	$0.8D$
Prevalence	$\forall k$	Counts instantiations of relation k	$\geq$	$0.8D$
Separation (ind)	$\forall w$	Counts times w selected as i	$\leq$	2
Separation (arg)	$\forall w$	Counts times w selected as part of a	$\leq$	1

Table 1: Each constraint takes the form $\mathbb{E}_q[f(z, a, i)] \leq b$ or $\mathbb{E}_q[f(z, a, i)] \geq b$; D denotes the number of corpus documents, $\forall k$ means one constraint per relation type, and $\forall w$ means one constraint per token in the corpus.

type separately generate a copy of the corpus, the $\hat{\theta}$ updates becomes closed-form expressions similar to equation 6. This approximation yields similar parameter estimates as the true updates while vastly improving speed, so we use it in our experiments.

5 Declarative Constraints

We now have the machinery to incorporate a variety of declarative constraints during inference. The classes of domain-independent constraints we study are summarized in Table 1. For the proportion constraints we arbitrarily select a threshold of 80% without any tuning, in the spirit of building a domain-independent approach.

Syntax As previous work has observed, most relations are expressed using a limited number of common syntactic patterns (Riloff, 1996; Banko and Etzioni, 2008). Our syntactic constraint captures this insight by requiring that a certain proportion of the induced instantiations for each relation match one of these syntactic patterns:

- The indicator is a verb and the argument's headword is either the child or grandchild of the indicator word in the dependency tree.
- The indicator is a noun and the argument is a modifier or complement.
- The indicator is a noun in a verb's subject and the argument is in the corresponding object.

Prevalence For a relation to be domain-relevant, it should occur in numerous documents across the corpus, so we institute a constraint on the number of times a relation is instantiated. Note that the effect of this constraint could also be achieved by tuning the prior probability of a relation not occurring in a document. However, this prior would need to be adjusted every time the number of documents or feature selection changes; using a constraint is an appealing alternative that is portable across domains.

Separation The separation constraint encourages diversity in the discovered relation types by restricting the number of times a single word can serve as either an indicator or part of the argument of a relation instance. Specifically, we require that every token of the corpus occurs at most once as a word in a relation's argument in expectation. On the other hand, a single word can sometimes be evocative of multiple relations (*e.g.*, "occurred" signals both *date* and *time* in "occurred on Friday at 3pm"). Thus, we allow each word to serve as an indicator more than once, arbitrarily fixing the limit at two.

6 Experimental Setup

Datasets and Metrics We evaluate on two datasets, financial market reports and newswire articles about earthquakes, previously used in work on high-level content analysis (Barzilay and Lee, 2004; Lapata, 2006). *Finance* articles chronicle daily market movements of currencies and stock indexes, and *earthquake* articles document specific earthquakes. Constituent parses are obtained automatically using the Stanford parser (Klein and Manning, 2003) and then converted to dependency parses using the PennConvertor tool (Johansson and Nugues, 2007). We manually annotated relations for both corpora, selecting relation types that occurred frequently in each domain. We found 15 types for *finance* and 9 for *earthquake*. Corpus statistics are summarized below, and example relation types are shown in Table 2.

	Docs	Sent/Doc	Tok/Doc	Vocab
Finance	100	12.1	262.9	2918
Earthquake	200	9.3	210.3	3155

In our task, annotation conventions for desired output relations can greatly impact token-level performance, and the model cannot learn to fit a particular convention by looking at example data. For example, earthquakes times are frequently reported in both local and GMT, and either may be arbitrar-

Finance	Bond	104.58 yen, 98.37 yen
	Dollar Change	up 0.52 yen, down 0.01 yen
	Tokyo Index Change	down 5.38 points or 0.41 percent, up 0.16 points, insignificant in percentage terms
Earthquake	Damage	about 10000 homes, some buildings, no information
	Epicenter	Patuca about 185 miles (300 kilometers) south of Quito, 110 kilometers (65 miles) from shore under the surface of the Flores sea in the Indonesian archipelago
	Magnitude	5.7, 6, magnitude-4

Table 2: Example relation types identified in the *finance* and *earthquake* datasets with example instance arguments.

ily chosen as correct. Moreover, the baseline we compare against produces lambda calculus formulas rather than spans of text as output, so a token-level comparison requires transforming its output.

For these reasons, we evaluate on both *sentence-level* and *token-level* precision, recall, and F-score. Precision is measured by mapping every induced relation cluster to its closest gold relation and computing the proportion of predicted sentences or words that are correct. Conversely, for recall we map every gold relation to its closest predicted relation and find the proportion of gold sentences or words that are predicted. This mapping technique is based on the many-to-one scheme used for evaluating unsupervised part-of-speech induction (Johnson, 2007). Note that sentence-level scores are always at least as high as token-level scores, since it is possible to select a sentence correctly but none of its true relation tokens while the opposite is not possible.

Domain-specific Constraints On top of the cross-domain constraints from Section 5, we study whether imposing basic domain-specific constraints can be beneficial. The *finance* dataset is heavily quantitative, so we consider applying a single domain-specific constraint stating that most relation arguments should include a number. Likewise, *earthquake* articles are typically written with a majority of the relevant information toward the beginning of the document, so its domain-specific constraint is that most relations should occur in the first two sentences of a document. Note that these domain-specific constraints are not specific to individual relations or instances, but rather encode a preference across all relation types. In both cases, we again use an 80% threshold without tuning.

Features For indicators, we use the word, part of speech, and word stem. For arguments, we use the word, syntactic constituent label, the head word of the parent constituent, and the dependency label of the argument to its parent.

Baselines We compare against three alternative unsupervised approaches. Note that the first two only identify relation-bearing sentences, not the specific words that participate in the relation.

Clustering (CLUTO): A straightforward way of identifying sentences bearing the same relation is to simply cluster them. We implement a clustering baseline using the CLUTO toolkit with word and part-of-speech features. As with our model, we set the number of clusters K to the true number of relation types.

Mallows Topic Model (MTM): Another technique for grouping similar sentences is the Mallows-based topic model of Chen et al. (2009). The datasets we consider here exhibit high-level regularities in content organization, so we expect that a topic model with global constraints could identify plausible clusters of relation-bearing sentences. Again, K is set to the true number of relation types.

Unsupervised Semantic Parsing (USP): Our final unsupervised comparison is to USP, an unsupervised deep semantic parser introduced by Poon and Domingos (2009). USP induces a lambda calculus representation of an entire corpus and was shown to be competitive with open information extraction approaches (Lin and Pantel, 2001; Banko et al., 2007). We give USP the required Stanford dependency format as input (de Marneffe and Manning, 2008). We find that the results are sensitive to the cluster granularity prior, so we tune this parameter and report the best-performing runs.

We recognize that USP targets a different output representation than ours: a hierarchical semantic structure over the entirety of a dependency-parsed text. In contrast, we focus on discovering a limited number K of domain-relevant relations expressed as

constituent phrases. Despite these differences, both methods ultimately aim to capture domain-specific relations expressed with varying verbalizations, and both operate over in-domain input corpora supplemented with syntactic information. For these reasons, USP provides a clear and valuable point of comparison. To facilitate this comparison, we transform USP's output lambda calculus formulas to relation spans as follows. First, we group lambda forms by a combination of core form, argument form, and the parent's core form.[7] We then filter to the K relations that appear in the most documents. For token-level evaluation we take the dependency tree fragment corresponding to the lambda form. For example, in the sentence "a strong earthquake rocked the Philippines island of Mindoro early Tuesday," USP learns that the word "Tuesday" has a core form corresponding to words {*Tuesday*, *Wednesday*, *Saturday*}, a parent form corresponding to words {*shook*, *rock*, *hit*, *jolt*}, and an argument form of TMOD; all phrases with this same combination are grouped as a relation.

Training Regimes and Hyperparameters For each run of our model we perform three random restarts to convergence and select the posterior with lowest final free energy. We fix K to the true number of annotated relation types for both our model and USP and L (the number of document segments) to five. Dirichlet hyperparameters are set to 0.1.

7 Results

Table 3's first two sections present the results of our main evaluation. For *earthquake*, the far more difficult domain, our base model with only the domain-independent constraints strongly outperforms all three baselines across both metrics. For *finance*, the CLUTO and USP baselines achieve performance comparable to or slightly better than our base model. Our approach, however, has the advantage of providing a formalism for seamlessly incorporating additional arbitrary domain-specific constraints. When we add such constraints (denoted as *model+DSC*), we achieve consistently higher performance than all baselines across both datasets and metrics, demonstrating that this approach provides a simple and effective framework for injecting domain knowledge into relation discovery.

The first two baselines correspond to a setup where the number of sentence clusters K is set to the true number of relation types. This has the effect of lowering precision because each sentence must be assigned a cluster. To mitigate this impact, we experimented with using $K + N$ clusters, with N ranging from 1 to 30. In each case, we then keep only the K largest clusters. For the *earthquake* dataset, increasing N improves performance until some point, after which performance degrades. However, the best F-Score corresponding to the optimal number of clusters is 42.2, still far below our model's 66.0 F-score. For the *finance* domain, increasing the number of clusters hurts performance.

Our results show a large gap in F-score between the sentence and token-level evaluations for both the USP baseline and our model. A qualitative analysis of the results indicates that our model often picks up on regularities that are difficult to distinguish without relation-specific supervision. For *earthquake*, a *location* may be annotated as "the Philippine island of Mindoro" while we predict just the word "Mindoro." For *finance*, an *index change* can be annotated as "30 points, or 0.8 percent," while our model identifies "30 points" and "0.8 percent" as separate relations. In practice, these outputs are all plausible discoveries, and a practitioner desiring specific outputs could impose additional constraints to guide relation discovery toward them.

The Impact of Constraints To understand the impact of the declarative constraints, we perform an ablation analysis on the constraint sets. We consider removing the constraints on syntactic patterns (*no-syn*) and the constraints disallowing relations to overlap (*no-sep*) from the full domain-independent model.[8] We also try a version with hard syntactic constraints (*hard-syn*), which requires that *every* extraction match one of the three syntactic patterns specified by the syntactic constraint.

Table 3's bottom section presents the results of this evaluation. The model's performance degrades when either of the two constraint sets are removed, demonstrating that the constraints are in fact benefi-

[7] This grouping mechanism yields better results than only grouping by core form.

[8] Prevalence constraints are always enforced, as otherwise the prior on not instantiating a relation would need to be tuned.

	Finance						Earthquake					
	Sentence-level			Token-level			Sentence-level			Token-level		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Model	82.1	59.7	69.2	42.2	23.9	30.5	54.2	68.1	60.4	20.2	16.8	18.3
Model+DSC	87.3	81.6	**84.4**	51.8	30.0	**38.0**	66.4	65.6	**66.0**	22.6	23.1	**22.8**
CLUTO	56.3	92.7	70.0	—	—	—	19.8	58.0	29.5	—	—	—
MTM	40.4	99.3	57.5	—	—	—	18.6	74.6	29.7	—	—	—
USP	91.3	66.1	76.7	28.5	32.6	30.4	61.2	43.5	50.8	9.9	32.3	15.1
No-sep	97.8	35.4	52.0	86.1	8.7	15.9	42.2	21.9	28.8	16.1	4.6	7.1
No-syn	83.3	46.1	59.3	20.8	9.9	13.4	53.8	60.9	57.1	14.0	13.8	13.9
Hard-syn	47.7	39.0	42.9	11.6	7.0	8.7	55.0	66.2	60.1	20.1	17.3	18.6

Table 3: Top section: our model, with and without domain-specific constraints (DSC). Middle section: The three baselines. Bottom section: ablation analysis of constraint sets for our model. For all scores, higher is better.

cial for relation discovery. Additionally, in the *hard-syn* case, performance drops dramatically for *finance* while remaining almost unchanged for *earthquake*. This suggests that formulating constraints as soft inequalities on posterior expectations gives our model the flexibility to accommodate both the underlying signal in the data and the declarative constraints.

Comparison against Supervised CRF Our final set of experiments compares a *semi-supervised* version of our model against a conditional random field (CRF) model. The CRF model was trained using the same features as our model's argument features. To incorporate training examples in our model, we simply treat annotated relation instances as observed variables. For both the baselines and our model, we experiment with using up to 10 annotated documents. At each of those levels of supervision, we average results over 10 randomly drawn training sets.

At the sentence level, our model compares very favorably to the supervised CRF. For *finance*, it takes at least 10 annotated documents (corresponding to roughly 130 individually annotated relation instances) for the CRF to match the semi-supervised model's performance. For *earthquake*, using even 10 annotated documents (about 71 relation instances) is not sufficient to match our model's performance.

At the token level, the supervised CRF baseline is far more competitive. Using a single labeled document (13 relation instances) yields superior performance to either of our model variants for *finance*, while four labeled documents (29 relation instances) do the same for *earthquake*. This result is not surprising—our model makes strong domain-independent assumptions about how underlying patterns of regularities in the text connect to relation expression. Without domain-specific supervision such assumptions are necessary, but they can prevent the model from fully utilizing available labeled instances. Moreover, being able to annotate even a single document requires a broad understanding of every relation type germane to the domain, which can be infeasible when there are many unfamiliar, complex domains to process.

In light of our strong sentence-level performance, this suggests a possible human-assisted application: use our model to identify promising relation-bearing sentences in a new domain, then have a human annotate those sentences for use by a supervised approach to achieve optimal token-level extraction.

8 Conclusions

This paper has presented a constraint-based approach to in-domain relation discovery. We have shown that a generative model augmented with declarative constraints on the model posterior can successfully identify domain-relevant relations and their instantiations. Furthermore, we found that a single set of constraints can be used across divergent domains, and that tailoring constraints specific to a domain can yield further performance benefits.

Acknowledgements

The authors gratefully acknowledge the support of Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0172. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the DARPA, AFRL, or the US government. Thanks also to Hoifung Poon and the members of the MIT NLP group for their suggestions and comments.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of DL*.

Michele Banko and Oren Etzioni. 2008. The tradeoffs between open and traditional relation extraction. In *Proceedings of ACL*.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of IJCAI*.

Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of HLT/NAACL*.

Kedar Bellare and Andrew McCallum. 2009. Generalized expectation criteria for bootstrapping extractors using record-text alignment. In *Proceedings of EMNLP*.

Jordan Boyd-Graber and David M. Blei. 2008. Syntactic topic models. In *Advances in NIPS*.

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

Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208.

Ming-Wei Chang, Lev Ratinov, and Dan Roth. 2007. Guiding semi-supervision with constraint-driven learning. In *Proceedings of ACL*.

Chaitanya Chemudugunta, Padhraic Smyth, and Mark Steyvers. 2006. Modeling general and specific aspects of documents with a probabilistic topic model. In *Advances in NIPS*.

Jinxiu Chen, Dong-Hong Ji, Chew Lim Tan, and Zheng-Yu Niu. 2005. Automatic relation extraction with model order selection and discriminative label identification. In *Proceedings of IJCNLP*.

Harr Chen, S.R.K. Branavan, Regina Barzilay, and David R. Karger. 2009. Content modeling using latent permutations. *JAIR*, 36:129–163.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The stanford typed dependencies representation. In *Proceedings of the Workshop on Cross-framework and Cross-domain Parser Evaluation*.

João Graça, Kuzman Ganchev, and Ben Taskar. 2007. Expectation maximization and posterior constraints. In *Advances in NIPS*.

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2004. Discovering relations among named entities from large corpora. In *Proceedings of ACL*.

Richard Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for english. In *Proceedings of NODALIDA*.

Mark Johnson. 2007. Why doesn't EM find good HMM POS-taggers? In *Proceedings of EMNLP*.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL*.

Mirella Lapata. 2006. Automatic evaluation of information ordering: Kendall's tau. *Computational Linguistics*, 32(4):471–484.

Dekang Lin and Patrick Pantel. 2001. DIRT - discovery of inference rules from text. In *Proceedings of SIGKDD*.

Gideon S. Mann and Andrew McCallum. 2008. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *Proceedings of ACL*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL/IJCNLP*.

Hoifung Poon and Pedro Domingos. 2009. Unsupervised semantic parsing. In *Proceedings of EMNLP*.

Ellen Riloff. 1996. Automatically generating extraction patterns from untagged texts. In *Proceedings of AAAI*.

Benjamin Rosenfeld and Ronen Feldman. 2007. Clustering for unsupervised relation identification. In *Proceedings of CIKM*.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of CoNLL*.

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of HLT/NAACL*.

Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. 2003. An improved extraction pattern representation model for automatic IE pattern acquisition. In *Proceedings of ACL*.

Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. 2000. Automatic acquisition of domain knowledge for information extraction. In *Proceedings of COLING*.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2010. Cross-document relation extraction without labelled data. In *Proceedings of EMNLP*.

Alexander Yates and Oren Etzioni. 2009. Unsupervised methods for determining object and relation synonyms on the web. *JAIR*, 34:255–296.

Min Zhang, Jian Su, Danmei Wang, Guodong Zhou, and Chew Lim Tan. 2005. Discovering relations between named entities from a large raw corpus using tree similarity-based clustering. In *IJCNLP*.

Jun Zhu, Zaiqing Nie, Xiaojing Liu, Bo Zhang, and Ji-Rong Wen. 2009. StatSnowball: a statistical approach to extracting entity relationships. In *Proceedings of WWW*.