Multi-instance Multi-label Learning for Relation Extraction

Mihai Surdeanu[†], Julie Tibshirani[†], Ramesh Nallapati[⋆], Christopher D. Manning[†]
[†] Stanford University, Stanford, CA 94305
{mihais,jtibs,manning}@stanford.edu
[⋆] Artificial Intelligence Center, SRI International
nallapat@ai.sri.com

Abstract

Distant supervision for relation extraction (RE) – gathering training data by aligning a database of facts with text – is an efficient approach to scale RE to thousands of different relations. However, this introduces a challenging learning scenario where the relation expressed by a pair of entities found in a sentence is unknown. For example, a sentence containing *Balzac* and *France* may express *BornIn* or *Died*, an unknown relation, or no relation at all. Because of this, traditional supervised learning, which assumes that each example is explicitly mapped to a label, is not appropriate. We propose a novel approach to multi-instance multi-label learning for RE, which jointly models all the instances of a pair of entities in text and all their labels using a graphical model with latent variables. Our model performs competitively on two difficult domains.

1 Introduction

Information extraction (IE), defined as the task of extracting structured information (e.g., events, binary relations, etc.) from free text, has received renewed interest in the "big data" era, when petabytes of natural-language text containing thousands of different structure types are readily available. However, traditional supervised methods are unlikely to scale in this context, as training data is either limited or nonexistent for most of these structures. One of the most promising approaches to IE that addresses this limitation is *distant supervision*, which generates training data automatically by aligning a database of facts with text (Craven and Kumlien, 1999; Bunescu and Mooney, 2007).

$$\text{DB} = \begin{pmatrix} \textit{BornIn}(\text{Barack Obama, United States}) \\ \textit{EmployedBy}(\text{Barack Obama, United States}) \end{pmatrix}$$

Sentence	Latent Label
Barack Obama is the 44th and current President of the United States.	*EmployedBy*
Obama was born in the United States just as he has always said.	*BornIn*
United States President Barack Obama meets with Chinese Vice President Xi Jinping today.	*EmployedBy*
Obama ran for the United States Senate in 2004.	–

Figure 1: Training sentences generated through distant supervision for a database containing two facts.

In this paper we focus on distant supervision for relation extraction (RE), a subproblem of IE that addresses the extraction of labeled relations between two named entities. Figure 1 shows a simple example for a RE domain with two labels. Distant supervision introduces two modeling challenges, which we highlight in the table. The first challenge is that some training examples obtained through this heuristic are not valid, e.g., the last sentence in Figure 1 is not a correct example for any of the known labels for the tuple. The percentage of such false positives can be quite high. For example, Riedel et al. (2010) report up to 31% of false positives in a corpus that matches Freebase relations with New York Times articles. The second challenge is that the same pair of entities may have multiple labels and it is unclear which label is instantiated by any textual mention of the given tuple. For example, in Figure 1, the tuple (*Barack Obama*, *United States*) has two valid labels: *BornIn* and *EmployedBy*, each (latently) instantiated in different sentences. In the

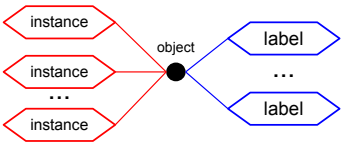

Figure 2: Overview of multi-instance multi-label learning. To contrast, in traditional supervised learning there is one instance and one label per object. For relation extraction the object is a tuple of two named entities. Each mention of this tuple in text generates a different instance.

Riedel corpus, 7.5% of the entity tuples in the training partition have more than one label.

We summarize this multi-instance multi-label (MIML) learning problem in Figure 2. In this paper we propose a novel graphical model, which we called MIML-RE, that targets MIML learning for relation extraction. Our work makes the following contributions:

(a) To our knowledge, MIML-RE is the first RE approach that jointly models both multiple instances (by modeling the latent labels assigned to instances) and multiple labels (by providing a simple method to capture dependencies between labels). For example, our model learns that certain labels tend to be generated jointly while others cannot be jointly assigned to the same tuple.

(b) We show that MIML-RE performs competitively on two difficult domains.

2 Related Work

Distant supervision for IE was introduced by Craven and Kumlien (1999), who focused on the extraction of binary relations between proteins and cells/tissues/diseases/drugs using the Yeast Protein Database as a source of distant supervision. Since then, the approach grew in popularity (Bunescu and Mooney, 2007; Bellare and McCallum, 2007; Wu and Weld, 2007; Mintz et al., 2009; Riedel et al., 2010; Hoffmann et al., 2011; Nguyen and Moschitti, 2011; Sun et al., 2011; Surdeanu et al., 2011a). However, most of these approaches make one or more approximations in learning. For example, most proposals heuristically transform distant supervision to traditional supervised learning (i.e., single-instance single-label) (Bellare and McCallum, 2007; Wu and Weld, 2007; Mintz et al., 2009; Nguyen and Moschitti, 2011; Sun et al., 2011; Surdeanu et al., 2011a). Bunescu and Mooney (2007) and Riedel et al. (2010) model distant supervision for relation extraction as a multi-instance single-label problem, which allows multiple mentions for the same tuple but disallows more than one label per object. Our work is closest to Hoffmann et al. (2011). They address the same problem we do (binary relation extraction) with a MIML model, but they make two approximations. First, they use a deterministic model that aggregates latent instance labels into a set of labels for the corresponding tuple by OR-ing the classification results. We use instead an object-level classifier that is trained jointly with the classifier that assigns latent labels to instances and can capture dependencies between labels. Second, they use a Perceptron-style additive parameter update approach, whereas we train in a Bayesian framework. We show in Section 5 that these approximations generally have a negative impact on performance.

MIML learning has been used in fields other than natural language processing. For example, Zhou and Zhang (2007) use MIML for scene classification. In this problem, each image may be assigned multiple labels corresponding to the different scenes captured. Furthermore, each image contains a set of patches, which forms the bag of instances assigned to the given object (image). Zhou and Zhang propose two algorithms that reduce the MIML problem to a more traditional supervised learning task. In one algorithm, for example, they convert the task to a multi-instance single-label problem by creating a separate bag for each label. Due to this, the proposed approach cannot model inter-label dependencies. Moreover, the authors make a series of approximations, e.g., they assume that each instance in a bag shares the bag's overall label. We instead model all these issues explicitly in our approach.

In general, our approach belongs to the category of models that learn in the presence of incomplete or incorrect labels. There has been interest among machine learning researchers in the general problem of noisy data, especially in the area of instance-based learning. Brodley and Friedl (1999) summarize past approaches and present a simple, all-purpose method to filter out incorrect data before training. While potentially applicable to our problem, this approach is completely general and cannot incorporate our domain-specific knowledge about how the noisy

data is generated.

3 Distant Supervision for Relation Extraction

Here we focus on distant supervision for the extraction of *relations between two entities*. We define a *relation* as the construct $r(e_1, e_2)$, where r is the relation name, e.g., *BornIn* in Figure 1, and e_1 and e_2 are two entity names, e.g., *Barack Obama* and *United States*. Note that there are entity tuples (e_1, e_2) that participate in multiple relations, $r_1, \ldots, r_i$. In other words, the tuple (e_1, e_2) is the object illustrated in Figure 2 and the different relation names are the labels. We define an *entity mention* as a sequence of text tokens that matches the corresponding entity name in some text, and *relation mention* (for a given relation $r(e_1, e_2)$) as a pair of entity mentions of e_1 and e_2 in the same sentence. Relation mentions thus correspond to the instances in Figure 2.[1] As the latter definition indicates, we focus on the extraction of relations expressed in a single sentence. Furthermore, we assume that entity mentions are extracted by a different process, such as a named entity recognizer.

We define the task of *relation extraction* as a function that takes as input a document collection ($\mathcal{C}$), a set of entity mentions extracted from $\mathcal{C}$ ($\mathcal{E}$), a set of known relation labels ($\mathcal{L}$) and an extraction model, and outputs a set of relations ($\mathcal{R}$) such that any of the relations extracted is supported by at least one sentence in $\mathcal{C}$. To train the extraction model, we use a database of relations ($\mathcal{D}$) that are instantiated at least once in $\mathcal{C}$. Using distant supervision, $\mathcal{D}$ is aligned with sentences in $\mathcal{C}$, producing relation mentions for all relations in $\mathcal{D}$.

4 Model

Our model assumes that each relation mention involving an entity pair has exactly one label, but allows the pair to exhibit multiple labels across different mentions. Since we do not know the actual relation label of a mention in the distantly supervised setting, we model it using a latent variable z that can take one of the k pre-specified relation labels as well as an additional *NIL* label, if no relation is expressed by the corresponding mention. We model the multiple relation labels an entity pair can assume

[1]For this reason, we use relation mention and relation instance interchangeably in this paper.

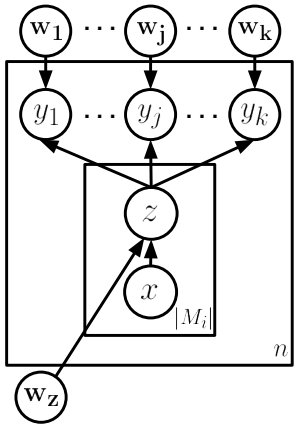

Figure 3: MIML model plate diagram. We unrolled the y plate to emphasize that it is a collection of binary classifiers (one per relation label), whereas the z classifier is multi-class. Each z and y_j classifier has an additional prior parameter, which is omitted here for clarity.

using a multi-label classifier that takes as input the latent relation types of the all the mentions involving that pair. The two-layer hierarchical model is shown graphically in Figure 3, and is described more formally below. The model includes one multi-class classifier (for z) and a set of binary classifiers (for each y_j). The z classifier assigns latent labels from $\mathcal{L}$ to individual relation mentions or *NIL* if no relation is expressed by the mention. Each y_j classifier decides if relation j holds for the given entity tuple, using the mention-level classifications as input. Specifically, in the figure:

- n is the number of distinct entity tuples in $\mathcal{D}$;
- M_i is the set of mentions for the ith entity pair;
- x is a sentence and z is the latent relation classification for that sentence;
- $\mathbf{w_z}$ is the weight vector for the multi-class mention-level classifier;
- k is the number of known relation labels in $\mathcal{L}$;
- y_j is the top-level classification decision for the entity pair as to whether the jth relation holds;
- $\mathbf{w_j}$ is the weight vector for the binary top-level classifier for the jth relation.

Additionally, we define P_i (N_i) as the set of all known positive (negative) relation labels for the ith entity tuple. In this paper, we construct N_i as $\mathcal{L} \setminus P_i$, but, in general, other scenarios are possible. For example, both Sun et al. (2011) and Surdeanu et

al. (2011a) proposed models where N_i for the ith tuple (e_1, e_2) is defined as: $\{r_j \mid r_j(e_1, e_k) \in \mathcal{D}, e_k \neq e_2, r_j \notin P_i\}$, which is a subset of $\mathcal{L} \setminus P_i$. That is, entity e_2 is considered a negative example for relation r_j (in the context of entity e_1) only if r_j exists in the training data with a different value.

The addition of the object-level layer (for $\mathbf{y}$) is an important contribution of this work. This layer can capture information that cannot be modeled by the mention-level classifier. For example, it can learn that two relation labels (e.g., *BornIn* and *SpouseOf*) cannot be generated jointly for the same entity tuple. So, if the z classifier outputs both these labels for different mentions of the same tuple, the $\mathbf{y}$ layer can cancel one of them. Furthermore, the $\mathbf{y}$ classifiers can learn when two labels tend to appear jointly, e.g., *CapitalOf* and *Contained* between two locations, and use this occurrence as positive reinforcement for these labels. We discuss the features that implement these ideas in Section 5.

4.1 Training

We train the proposed model using hard discriminative Expectation Maximization (EM). In the Expectation (E) step we assign latent mention labels using the current model (i.e., the mention and relation level classifiers). In the Maximization (M) step we retrain the model to maximize the log likelihood of the data using the current latent assignments.

In the equations that follow, we refer to $\mathbf{w_1}, \ldots, \mathbf{w_k}$ collectively as $\mathbf{w_y}$ for compactness. The vector $\mathbf{z_i}$ contains the latent mention-level classifications for the ith entity pair, while $\mathbf{y_i}$ represents the corresponding set of gold-standard labels (that is, $y_i^{(r)} = 1$ if $r \in P_i$, and $y_i^{(r)} = 0$ for $r \in N_i$.) Using these notations, the log-likelihood of the data is given by:

$$\begin{aligned} LL(\mathbf{w_y}, \mathbf{w_z}) &= \sum_{i=1}^{n} \log p(\mathbf{y_i}|\mathbf{x_i}, \mathbf{w_y}, \mathbf{w_z}) \\ &= \sum_{i=1}^{n} \log \sum_{\mathbf{z_i}} p(\mathbf{y_i}, \mathbf{z_i}|\mathbf{x_i}, \mathbf{w_y}, \mathbf{w_z}) \end{aligned}$$

The joint probability in the inner summation can be broken up into simpler parts:

$$\begin{aligned} &p(\mathbf{y_i}, \mathbf{z_i}|\mathbf{x_i}, \mathbf{w_y}, \mathbf{w_z}) \\ &\quad = p(\mathbf{z_i}|\mathbf{x_i}, \mathbf{w_z}) p(\mathbf{y_i}|\mathbf{z_i}, \mathbf{w_y}) \\ &\quad = \prod_{m \in M_i} p(z_i^{(m)}|x_i^{(m)}, \mathbf{w_z}) \prod_{r \in P_i \cup N_i} p(y_i^{(r)}|\mathbf{z_i}, \mathbf{w_y^{(r)}}) \end{aligned}$$

where the last step follows from conditional independence. Thus the log-likelihood for this problem is not convex (it includes a sum of products). However, we can still use EM, but the optimization focuses on maximizing the lower bound of the log-likelihood, i.e., we maximize the above joint probability for each entity pair in the database. Rewriting this probability in log space, we obtain:

$$\begin{aligned} &\log p(\mathbf{y_i}, \mathbf{z_i}|\mathbf{x_i}, \mathbf{w_y}, \mathbf{w_z}) \qquad (1) \\ &\qquad = \sum_{m \in M_i} \log p(z_i^{(m)}|x_i^{(m)}, \mathbf{w_z}) + \\ &\qquad \sum_{r \in P_i \cup N_i} \log p(y_i^{(r)}|\mathbf{z_i}, \mathbf{w_y^{(r)}}) \end{aligned}$$

The algorithm proceeds as follows.

E-step: In this step we infer the mention-level classifications $\mathbf{z_i}$ for each entity tuple, given all its mentions, the gold labels $\mathbf{y_i}$, and current model, i.e., $\mathbf{w_z}$ and $\mathbf{w_y}$ weights. Formally, we seek to find:

$$\mathbf{z_i}^* = \arg\max_{\mathbf{z}} p(\mathbf{z}|\mathbf{y_i}, \mathbf{x_i}, \mathbf{w_y}, \mathbf{w_z})$$

However it is computationally intractable to consider all vectors $\mathbf{z}$ as there is an exponential number of possible assignments, so we approximate and consider each mention separately. Concretely,

$$\begin{aligned} &p(z_i^{(m)}|\mathbf{y_i}, \mathbf{x_i}, \mathbf{w_y}, \mathbf{w_z}) \\ &\qquad \propto p(\mathbf{y_i}, z_i^{(m)}|\mathbf{x_i}, \mathbf{w_y}, \mathbf{w_z}) \\ &\qquad \approx p(z_i^{(m)}|x_i^{(m)}, \mathbf{w_z}) p(\mathbf{y_i}|\mathbf{z_i'}, \mathbf{w_y}) \\ &\qquad = p(z_i^{(m)}|x_i^{(m)}, \mathbf{w_z}) \prod_{r \in P_i \cup N_i} p(y_i^{(r)}|\mathbf{z_i'}, \mathbf{w_y^{(r)}}) \end{aligned}$$

where $\mathbf{z_i'}$ contains the previously inferred mention labels for group i, with the exception of component m whose label is replaced by $z_i^{(m)}$. So for $i = 1, \ldots, n$, and for each $m \in M_i$ we calculate:

$$\begin{aligned} z_i^{(m)*} = \arg\max_{z} \; & p(z|x_i^{(m)}, \mathbf{w_z}) \times \qquad (2) \\ & \prod_{r \in P_i \cup N_i} p(y_i^{(r)}|\mathbf{z_i'}, \mathbf{w_y^{(r)}}) \end{aligned}$$

Intuitively, the above equation indicates that mention labels are chosen to maximize: (a) the probabilities assigned by the mention-level model; (b) the probability that the correct relation labels are assigned to the corresponding tuple; and (c) the probability that the labels known to be incorrect are *not* assigned to the tuple. For example, if a particular mention label receives a high mention-level probability but it is known to be a negative label for that tuple, it will receive a low overall score.

M-step: In this step we find $\mathbf{w_y}, \mathbf{w_z}$ that maximize the lower bound of the log-likelihood, i.e., the probability in equation (1), given the current assignments for $\mathbf{z_i}$. From equation (1) it is clear that this can be maximized separately with respect to $\mathbf{w_y}$ and $\mathbf{w_z}$. Intuitively, this step amounts to learning the weights for the mention-level classifier ($\mathbf{w_z}$) and the weights for each of the k top-level classifiers ($\mathbf{w_y}$). The updates are given by:

$$\mathbf{w_z^*} = \arg\max_{\mathbf{w}} \sum_{i=1}^{n} \sum_{m \in M_i} \log p(z_i^{(m)*} | x_i^{(m)}, \mathbf{w}) \quad (3)$$

$$\mathbf{w_y^{(r)*}} = \arg\max_{\mathbf{w}} \sum_{1 \le i \le n \ s.t. \ r \in P_i \cup N_i} \log p(y_i^{(r)} | \mathbf{z_i^*}, \mathbf{w}) \quad (4)$$

Note that these are standard updates for logistic regression. We obtained these weights using $k+1$ logistic classifiers: one multi-class classifier for $\mathbf{w_z}$ and k binary classifiers for each relation label $r \in \mathcal{L}$. We implemented all using the L2-regularized logistic regression from the publicly-downloadable Stanford CoreNLP package.[2] The main difference between the classifiers is how features are generated: the mention-level classifier computes its features based on x_i, whereas the relation-level classifiers generate features based on the current assignments for $\mathbf{z_i}$ and the corresponding relation label r. We discuss the actual features used in our experiments in Section 5.

4.2 Inference

Given an entity tuple, we obtain its relation labels as follows. We first classify its mentions:

$$z_i^{(m)*} = \arg\max_{z} p(z | x_i^{(m)}, \mathbf{w_z}) \quad (5)$$

then decide on the final relation labels using the top-level classifiers:

$$y_i^{(r)*} = \arg\max_{y \in \{0,1\}} p(y | \mathbf{z_i^*}, \mathbf{w_y^{(r)}}) \quad (6)$$

4.3 Implementation Details

We discuss next several details that are crucial for the correct implementation of the above model.

Initialization: Since EM is not guaranteed to converge at the global maximum of the observed data likelihood, it is important to provide it with good starting values. In our context, the initial values are labels assigned to $\mathbf{z_i}$, which are required to compute equation (2) in the first iteration ($\mathbf{z_i'}$). We generate these values using a local logistic regression classifier that uses the same features as the mention-level classifier in the joint model but treats each relation mention independently. We train this classifier using "traditional" distant supervision: for each relation in the database $\mathcal{D}$ we assume that all the corresponding mentions are positive examples for the corresponding label (Mintz et al., 2009). Note that this heuristic repeats relation mentions with different labels for the tuples that participate in multiple relations. For example, all the relation mentions in Figure 1 will yield datums with both the *EmployedBy* and *BornIn* labels. Despite this limitation, we found that this is a better initialization heuristic than random assignment.

For the second part of equation (2), we initialize the relation-level classifier with a model that replicates the *at least one* heuristic of Hoffmann et al. (2011). Each $\mathbf{w_y^{(r)}}$ model has a single feature with a high positive weight that is triggered when label r is assigned to any of the mentions in $\mathbf{z_i^*}$.

Avoiding overfitting: A naïve implementation of our approach leads to an unrealistic training scenario where the z classifier generates predictions (in equation (2)) for the same datums it has seen in training in the previous iteration. To avoid this overfitting problem we used cross validation: we divided the training tuples in K distinct folds and trained K different mention-level classifiers. Each classifier outputs $p(z | x_i^{(m)}, \mathbf{w_z})$ for tuples in a given fold during the E-step (equation (2)) and is trained (equation (3)) using tuples from all other folds.

[2] nlp.stanford.edu/software/corenlp.shtml

At testing time, we compute $p(z|x_i^{(m)}, \mathbf{w_z})$ in equation (5) as the average of the probabilities of the above set of mention classifiers:

$$p(z|x_i^{(m)}, \mathbf{w_z}) = \frac{\sum_{j=1}^{K} p(z|x_i^{(m)}, \mathbf{w_z^j})}{K}$$

where $\mathbf{w_z^j}$ are the weights of the mention classifier responsible for fold j. We found that this simple bagging model performs slightly better in practice (a couple of tenths of a percent) than training a single mention classifier on the latent mention labels generated in the last training iteration.

Inference during training: During the inference process in the E-step, the algorithm incrementally "flips" mention labels based on equation (2), for each group of mentions M_i. Thus, $\mathbf{z'_i}$ changes as the algorithm progresses, which may impact the label assigned to the remaining mentions in that group. To avoid any potential bias introduced by the arbitrary order of mentions as seen in the data, we randomize each group M_i before we inspect its mentions.

5 Experimental Results

5.1 Data

We evaluate our algorithm on two corpora. The first was developed by Riedel et al. (2010) by aligning Freebase[3] relations with the New York Times (NYT) corpus. They used the Stanford named entity recognizer (Finkel et al., 2005) to find entity mentions in text and constructed relation mentions only between entity mentions in the same sentence.

Riedel et al. (2010) observes that evaluating on this corpus underestimates true extraction accuracy because Freebase is incomplete. Thus, some relations extracted during testing will be incorrectly marked as wrong, simply because Freebase has no information on them. To mitigate this issue, Riedel et al. (2010) and Hoffman et al. (2011) perform a second evaluation where they compute the accuracy of labels assigned to a set of relation mentions that they manually annotated. To avoid any potential annotation biases, we instead evaluate on a second corpus that has comprehensive annotations generated by experts for all test relations.

We constructed this second dataset using mainly resources distributed for the 2010 and 2011 KBP shared tasks (Ji et al., 2010; Ji et al., 2011). We generated training relations from the knowledge base provided by the task organizers, which is a subset of the English Wikipedia infoboxes from a 2008 snapshot. Similarly to the corpus of Riedel et al., these infoboxes contain open-domain relations between named entities, but with a different focus. For example, more than half of the relations in the evaluation data are alternate names of organizations or persons (e.g., *org:alternate_names*) or relations associated with employment and membership (e.g., *per:employee_of*) (Ji et al., 2011). We aligned these relations against a document collection that merges two distinct sources: (a) the collection provided by the shared task, which contains approximately 1.5 million documents from a variety of sources, including newswire, blogs and telephone conversation transcripts; and (b) a complete snapshot of the English Wikipedia from June 2010. During training, for each entity tuple (e_1, e_2), we retrieved up to 50 sentences that contain both entity mentions.[4] We used Stanford's CoreNLP package to find entity mentions in text and, similarly to Riedel et al. (2010), we construct relation mention candidates only between entity mentions in the same sentence. We analyzed a set of over 2,000 relation mentions and we found that 39% of the mentions where e_1 is an organization name and 36% of mentions where e_1 is a person name do not express the corresponding relation.

At evaluation time, the KBP shared task requires the extraction of all relations $r(e_1, e_2)$ given a query that contains only the first entity e_1. To accommodate this setup, we adjusted our sentence extraction component to use just e_1 as the retrieval query and we kept up to 50 sentences that contain a mention of the input entity for each evaluation query. For tuning and testing we used the 200 queries from the 2010 and 2011 evaluations. We randomly selected 40 queries for development and used the remaining 160 for the formal evaluation.

To address the large number of negative examples in training, Riedel et al. subsampled them randomly with a retention probability of 10%. For the KBP corpus, we followed the same strategy, but we used

[3] `freebase.com`

[4] Sentences were ranked using the similarity between their parent document and the query that concatenates the two entity names. We used the default Lucene similarity measure.

	# of gold relations in training	# of gold relations in testing	% of gold entity tuples with more than one label in training	% of gold entity tuples with multiple mentions in text in training	% of mentions that do not express their relation	# of relation labels
Riedel	4,700	1,950	7.5%	46.4%	up to 31%	51
KBP	183,062	3,334	2.8%	65.1%	up to 39%	41

Table 1: Statistics about the two corpora used in this paper. Some of the numbers for the Riedel dataset is from (Riedel et al., 2010; Hoffmann et al., 2011).

a subsampling probability of 5% because this led to the best results in development for all models.

Table 1 provides additional statistics about the two corpora. The table indicates that having multiple mentions for an entity tuple is a very common phenomenon in both corpora, and that having multiple labels per tuple is more common in the Riedel dataset than KBP (7.5% vs. 2.8%).

5.2 Features

Our model requires two sets of features: one for the mention classifier (z) and one for the relation classifier (y). In the Riedel dataset, we used the same features as Riedel et al. (2010) and Hoffmann et al. (2011) for the mention classifier. In the KBP dataset, we used a feature set that was developed in our previous work (Surdeanu et al., 2011b). These features can be grouped in three classes: (a) features that model the two entities, such as their head words; (b) features that model the syntactic context of the relation mention, such as the dependency path between the two entity mentions; and (c) features that model the surface context, such as the sequence of part of speech tags between the two entity mentions. We used these features for all the models evaluated on the KBP dataset.[5]

For the relation-level classifier, we developed two feature groups. The first models Hoffmann et al.'s *at least one* heuristic using a single feature, which is set to true if at least one mention in $\mathbf{z_i}$ has the label r, which is modeled by the current relation classifier. The second group models the dependencies between relation labels. This is implemented by a set of $|\mathcal{L}| - 1$ features, where feature j is instantiated whenever the label modeled (r) is predicted jointly with another label r_j ($r_j \in \mathcal{L}, r_j \neq r$) in $\mathbf{z_i}$. These features learn both positive and negative reinforcements between labels. For example, if labels r_1 and r_2 tend to be generated jointly, the feature for the corresponding dependency will receive a positive weight in the models for r_1 and r_2. Similarly, if r_1 and r_2 cannot be generated jointly, the model will assign a negative weight to feature 2 in r_1's classifier and to feature 1 in r_2's classifier. Note that this feature is asymmetric, i.e., feature 1 in r_2's classifier may have a different value than feature 2 in r_1's classifier, depending on the accuracy of the individual predictions for r_1 and r_2.

5.3 Baselines

We compare our approach against three models:

Mintz++ – This is the model used to initialize the mention-level classifier in our model. As discussed in Section 4.3, this model follows the "traditional" distant supervision heuristic, similarly to (Mintz et al., 2009). However, our implementation has several advantages over the original model: (a) we model each relation mention independently, whereas Mintz et al. collapsed all the mentions of the same entity tuple into a single datum; (b) we allow multi-label outputs for a given entity tuple at prediction time by OR-ing the predictions for the individual relation mentions corresponding to the tuple (similarly to (Hoffmann et al., 2011))[6]; and (c) we use the simple bagging strategy described in Section 4.3 to combine multiple models. Empirically, we observed that these changes yield a significant improvement over the original proposal. For this reason, we consider this model a strong baseline on its own.

Riedel – This is the "at-least-once" model reported in (Riedel et al., 2010), which had the best performance in that work. This approach models the task as a multi-instance single-label problem. Note that this is the only model shown here that does not allow multi-label outputs for an entity tuple.

[5] To avoid an excessive number of features in the KBP experiments, we removed features seen less than five times in training.

[6] We also allow multiple labels per tuple at training time, in which case we replicate the corresponding datum for each label. However, this did not improve performance significantly compared to selecting a single label per datum during training.

Hoffmann – This is the "MultiR" model, which performed the best in (Hoffmann et al., 2011). This models RE as a MIML problem, but learns using a Perceptron algorithm and uses a deterministic "at least one" decision instead of a relation classifier. We used Hoffman's publicly released code[7] for the experiments on the Riedel dataset and our own implementation for the KBP experiments.[8]

5.4 Results

We tuned all models using three-fold cross validation for the Riedel dataset and using the development queries for the KBP dataset. `MIML-RE` has two parameters that require tuning: the number of EM epochs (T) and the number of folds for the mention classifiers (K).[9] The values obtained after tuning are $T = 15, K = 5$ for the Riedel dataset and $T = 8, K = 3$ for KBP. Similarly, we tuned the number of epochs for the Hoffmann model on the KBP dataset, obtaining an optimal value of 20.

On the Riedel dataset we evaluate all models using standard precision and recall measures. For the KBP evaluation we used the official KBP scorer,[10] with two changes: (a) we score with the parameter `anydoc` set to true, which configures the scorer to accept relation mentions as correct regardless of their supporting document; and (b) we score only on the subset of gold relations that have at least one mention in our sentences. The first decision is necessary because the gold KBP answers contain supporting documents only from the corpus provided by the organizers but we retrieve candidate answers from multiple collections. The second is required because the focus of this work is not on sentence retrieval but on RE, which should be evaluated in isolation.[11]

Similarly to previous work, we report precision/recall curves in Figure 4. We evaluate two variants of `MIML-RE`: one that includes all the features for the y model, and another (`MIML-RE At-Least-One`) which has only the *at least one* feature. For all the Bayesian models implemented here, we sorted the predicted relations by the noisy-or score of the top predictions for their mentions. Formally, we rank a relation r predicted for group i, i.e., $r \in \mathbf{y_i^*}$, using:

$$\text{noisyOr}_i(r) = 1 - \prod_{m \in M_i} (1 - s_i^{(m)}(r))$$

where $s_i^{(m)}(r) = p(r|x_i^{(m)}, \mathbf{w_z})$ if $r = z_i^{(m)*}$ or 0 otherwise. The noisy-or formula performs well for ranking because it integrates model confidence (the higher the probabilities, the higher the score) and redundancy (the more mentions are predicted with a label, the higher that label's score). Note that the above ranking score does not include the probability of the relation classifier (equation (6)) for `MIML-RE`. While we use equation (6) to generate $\mathbf{y_i^*}$, we found that the corresponding probabilities are too coarse to provide a good ranking score. This is caused by the fact that our relation-level classifier works with a small number of (noisy) features. Lastly, for our implementation of the Hoffmann et al. model, we used their ranking heuristic (sorting predictions by the maximum extraction score for that relation).

6 Discussion

Figure 4 indicates that `MIML-RE` generally outperforms the current state of the art. In the Riedel dataset, `MIML-RE` has higher overall recall than the Riedel et al. model, and, for the same recall point, `MIML-RE`'s precision is between 2 and 15 points higher. For most of the curve, our model obtains better precision for the same recall point than the Hoffmann model, which currently has the best reported results on this dataset. The difference is as high as 5 precision points around the middle of the curve. The Hoffmann model performs better close to the extremities of the curve (low/high recall). Nevertheless, we argue that our model is more stable than Hoffmann's: `MIML-RE` yields a smoother precision/recall curve, without most of the depressions seen in the Hoffmann results. In the KBP dataset, `MIML-RE` performs consistently better than our implementation of Hoffmann's model, with higher precision values for the same recall point, and much higher overall recall. We believe that these differences are caused by our Bayesian framework,

[7] `cs.washington.edu/homes/raphaelh/mr/`

[8] The decision to reimplement the Hoffmann model was a practical one, driven by incompatibilities between their implementation and our KBP framework.

[9] We could also tune the prior parameters for both our model and Mintz++, but we found in early experiments that the default value of 1 yields the best scores for all priors.

[10] `nlp.cs.qc.cuny.edu/kbp/2011/scoring.html`

[11] Due to these changes, the scores reported in this paper are not directly comparable with the shared task scores.

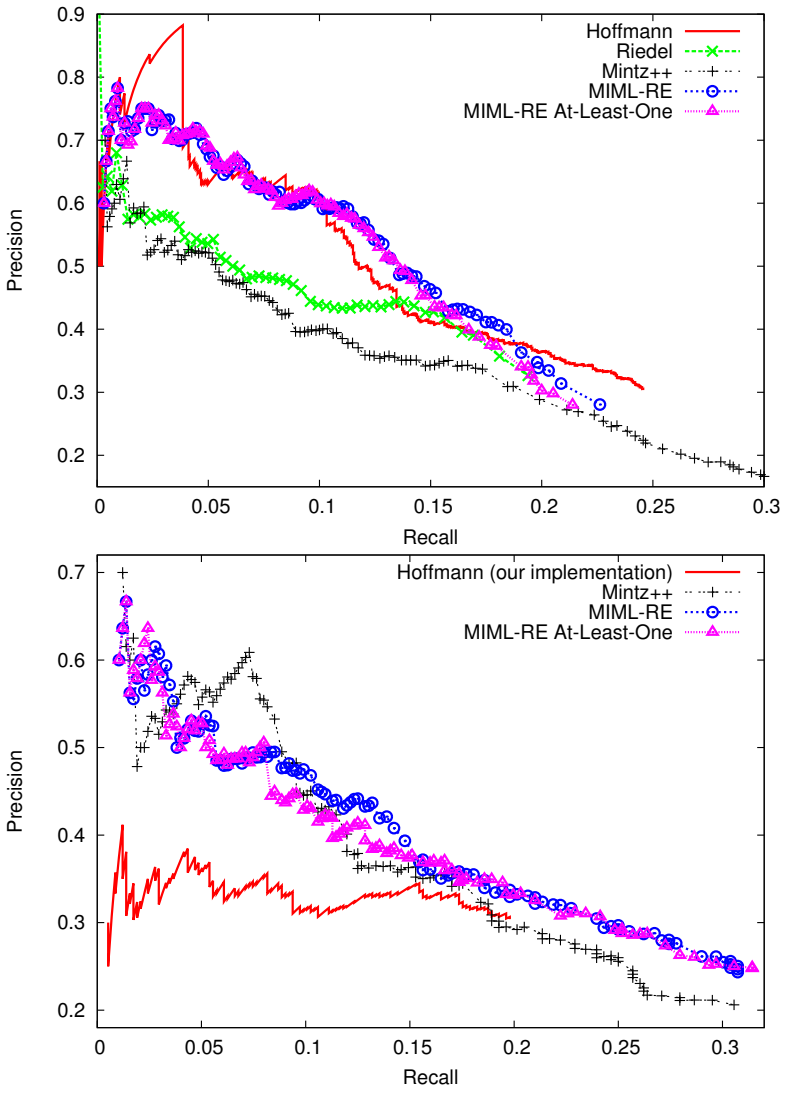

Figure 4: Results in the Riedel dataset (top) and the KBP dataset (bottom). The Hoffmann scores in the KBP dataset were generated using our implementation. The other Hoffmann and Riedel results were taken from their papers.

which provides a more formal implementation of the MIML problem.

Figure 4 also indicates that MIML-RE yields a consistent improvement over Mintz++ (with the exception of a few points in the low-recall portion of the KBP curves). The difference in precision for the same recall point is as high as 25 precision points in the Riedel dataset and up to 5 points in KBP. Overall, the best F1 score of MIML-RE is slightly over 1 point higher than the best F1 score of Mintz++ in the Riedel dataset and 3 points higher in KBP. Considering that Mintz++ is a strong baseline and we evaluate on two challenging domains, we consider these results proof that the correct modeling of the MIML scenario is beneficial.

Lastly, Figure 4 shows that MIML-RE outperforms its variant without label-dependency features (MIML-RE At-Least-One) in the higher-recall part of the curve in the Riedel dataset. The improvement is approximately 1 F1 point throughout the last segment of the curve. The overall increase in F1 was found to be significant ($p = 0.0296$) in a one-sided, paired t-test over randomly sampled test data. We see a smaller improvement in KBP (concentrated around the middle of the curve), likely because the number of entity tuples with multiple labels in training is small (see Table 1). Nevertheless, this exercise shows that, when dependencies between labels exist in a dataset, modeling them, which can be trivially done in MIML-RE, is useful.

	P	R	F1
Hoffmann (our implementation)	48.6	29.8	37.0
Mintz++	43.8	**36.8**	40.0
MIML-RE	**64.8**	31.6	**42.6**
MIML-RE At-Least-One	56.1	32.5	41.1

Table 2: Results at the highest F1 point in the precision/recall curve on the dataset that contains groups with at least 10 mentions.

In a similar vein, we tested the models previously described on a subset of the Riedel evaluation dataset that only includes groups with at least 10 mentions. This corpus contains approximately 2% of the groups from the original testing partition, out of which 90 tuples have at least one known label and 1410 groups serve as negative examples.

For conciseness, we do not include the entire precision/recall curves for this experiment, but summarize them in Table 2, which lists the performance peak (highest F1 score) for each of the models investigated. The table shows that `MIML-RE` obtains the highest F1 score overall, 1.5 points higher than `MIML-RE At-Least-One` and 2.6 points higher than Mintz++. More importantly, for approximately the same recall point, `MIML-RE` obtains a precision that is over 8 percentage points higher than that of `MIML-RE At-Least-One`. A post-hoc inspection of the results indicates that, indeed, `MIML-RE` successfully eliminates undesired labels when two (or more) incompatible labels are jointly assigned to the same tuple. Take for example the tuple (*Mexico City*, *Mexico*), for which the correct relation is */location/administrative_division/country*. `MIML-RE At-Least-One` incorrectly predicts the additional */location/location/contains* relation, while `MIML-RE` does not make this prediction because it recognizes that these two labels are incompatible in general: one location cannot both be within another location and contain it. Indeed, examining the weights assigned to label-dependency features in `MIML-RE`, we see that the model has assigned a large negative weight to the dependency feature between */location/location/contains* and */location/administrative_division/country* for the */location/location/contains* class. We also observe positive dependencies between labels. For example, `MIML-RE` learns that the relations */people/person/place_lived* and */people/person/place_of_birth* tend to co-occur and assigns a positive weight to this dependency feature for the corresponding classes.

These results strongly suggest that when all aspects of the MIML scenario are present, our model can successfully capture them and make use of the additional structure to improve performance.

7 Conclusion

In this paper we showed that distant supervision for RE, which generates training data by aligning a database of facts with text, poses a distinct multi-instance multi-label learning scenario. In this setting, each entity pair to be modeled typically has multiple instances in the text and may have multiple labels in the database. This is considerably different from traditional supervised learning, where each instance has a single, explicit label.

We argued that this MIML scenario should be formally addressed. We proposed, to our knowledge, the first approach that models all aspects of the MIML setting, i.e., the latent assignment of labels to instances and dependencies between labels assigned to the same entity pair.

We evaluated our model on two challenging domains and obtained state-of-the-art results on both. Our model performs well even when not all aspects of the MIML scenario are common, and as seen in the discussion, shows significant improvement when evaluated on entity pairs with many labels or mentions. When all aspects of the MIML scenario are present, our model is well-equipped to handle them.

The code and data used in the experiments reported in this paper are available at: `http://nlp.stanford.edu/software/mimlre.shtml`.

Acknowledgments

We 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-0181. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of the DARPA, AFRL, or the US government. We gratefully thank Raphael Hoffmann and Sebastian Riedel for sharing their code and data and for the many useful discussions.

References

Kedar Bellare and Andrew McCallum. 2007. Learning extractors from unlabeled text using relevant databases. In *Proceedings of the Sixth International Workshop on Information Extraction on the Web*.

Carla Brodley and Mark Friedl. 1999. Identifying mislabeled training data. *Journal of Artificial Intelligence Research (JAIR)*.

Razvan Bunescu and Raymond Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.

Mark Craven and Johan Kumlien. 1999. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*.

Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics*.

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 the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Heng Ji, Ralph Grishman, Hoa T. Dang, Kira Griffitt, and Joe Ellis. 2010. Overview of the TAC 2010 knowledge base population track. In *Proceedings of the Text Analytics Conference*.

Heng Ji, Ralph Grishman, and Hoa T. Dang. 2011. Overview of the TAC 2011 knowledge base population track. In *Proceedings of the Text Analytics Conference*.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics*.

Truc Vien T. Nguyen and Alessandro Moschitti. 2011. End-to-end relation extraction using distant supervision from external semantic repositories. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD '10)*.

Ang Sun, Ralph Grishman, Wei Xu, and Bonan Min. 2011. New York University 2011 system for KBP slot filling. In *Proceedings of the Text Analytics Conference*.

Mihai Surdeanu, Sonal Gupta, John Bauer, David McClosky, Angel X. Chang, Valentin I. Spitkovsky, and Christopher D. Manning. 2011a. Stanford's distantly-supervised slot-filling system. In *Proceedings of the Text Analytics Conference*.

Mihai Surdeanu, David McClosky, Mason R. Smith, Andrey Gusev, and Christopher D. Manning. 2011b. Customizing an information extraction system to a new domain. In *Proceedings of the Workshop on Relational Models of Semantics*, Portland, Oregon, June.

Fei Wu and Dan Weld. 2007. Autonomously semantifying Wikipedia. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*.

Z.H. Zhou and M.L. Zhang. 2007. Multi-instance multi-label learning with application to scene classification. In *Advances in Neural Information Processing Systems (NIPS)*.