Deep Probabilistic Logic: A Unifying Framework for Indirect Supervision

Hai Wang[1]* **Hoifung Poon**[2]
[1] Toyota Technological Institute at Chicago, Chicago, Illinois, USA
[2] Microsoft Research, Redmond, WA, USA
haiwang@ttic.edu hoifung@microsoft.com

Abstract

Deep learning has emerged as a versatile tool for a wide range of NLP tasks, due to its superior capacity in representation learning. But its applicability is limited by the reliance on annotated examples, which are difficult to produce at scale. Indirect supervision has emerged as a promising direction to address this bottleneck, either by introducing labeling functions to automatically generate noisy examples from unlabeled text, or by imposing constraints over interdependent label decisions. A plethora of methods have been proposed, each with respective strengths and limitations. Probabilistic logic offers a unifying language to represent indirect supervision, but end-to-end modeling with probabilistic logic is often infeasible due to intractable inference and learning. In this paper, we propose deep probabilistic logic (DPL) as a general framework for indirect supervision, by composing probabilistic logic with deep learning. DPL models label decisions as latent variables, represents prior knowledge on their relations using weighted first-order logical formulas, and alternates between learning a deep neural network for the end task and refining uncertain formula weights for indirect supervision, using variational EM. This framework subsumes prior indirect supervision methods as special cases, and enables novel combination via infusion of rich domain and linguistic knowledge. Experiments on biomedical machine reading demonstrate the promise of this approach.

1 Introduction

Deep learning has proven successful in a wide range of NLP tasks (Bahdanau et al., 2014; Bengio et al., 2003; Clark and Manning, 2016; Hermann et al., 2015; Sutskever et al., 2014). The versatility stems from its capacity of learning a compact representation of complex input patterns (Goodfellow et al., 2016). However, success of deep learning is bounded by its reliance on labeled examples, which are expensive and time-consuming to produce. Indirect supervision has emerged as a promising direction for breaching the annotation bottleneck. A powerful paradigm is *joint inference* (Chang et al., 2007; Poon and Domingos, 2008; Druck et al., 2008; Ganchev et al., 2010), which leverages linguistic and domain knowledge to impose constraints over interdependent label decisions. More recently, another powerful paradigm, often loosely called *weak supervision*, has gained in popularity. The key idea is to introduce labeling functions to automatically generate (noisy) training examples from unlabeled text. *Distant supervision* is a prominent example that used existing knowledge bases for this purpose (Craven and Kumlien, 1999; Mintz et al., 2009). *Data programming* went further by soliciting labeling functions from domain experts (Ratner et al., 2016; Bach et al., 2017).

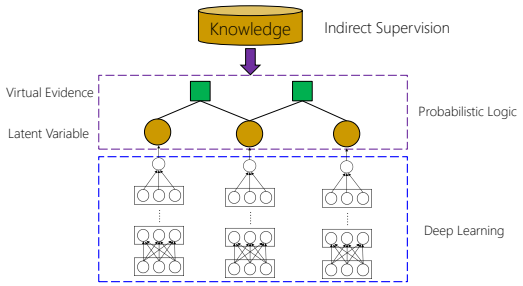


Figure 1: Deep Probabilistic Logic: A general framework for combining indirect supervision strategies by composing probabilistic logic with deep learning. Learning amounts to maximizing conditional likelihood of virtual evidence given input by summing up latent label decisions.

Indirect-supervision methods have achieved remarkable successes in a number of NLP tasks, but they also exhibit serious limitations. Distant supervision often produces incorrect labels, whereas labeling functions from data programming vary in

*This work was conducted at Microsoft Research.

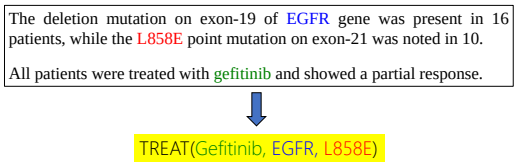


Figure 2: Example of cross-sentence relation extraction for precision cancer treatment.

quality and coverage, and may contradict with each other on individual instances. Joint inference incurs greater modeling complexity and often requires specialized learning and inference procedures.

Since these methods draw on diverse and often orthogonal sources of indirect supervision, combining them may help address their limitations and amplify their strengths. Probabilistic logic offers an expressive language for such an integration, and is well suited for resolving noisy and contradictory information (Richardson and Domingos, 2006). Unfortunately, probabilistic logic generally incurs intractable learning and inference, often rendering end-to-end modeling infeasible.

In this paper, we propose **deep probabilistic logic (DPL)** as a unifying framework for indirect supervision (Figure 1). Specifically, we made four contributions. First, we introduce a modular design to compose probabilistic logic with deep learning, with a supervision module that represents indirect supervision using probabilistic logic, and a prediction module that performs the end task using a deep neural network. Label decisions are modeled as latent variables and serve as the interface between the two modules.

Second, we show that all popular forms of indirect supervision can be represented in DPL by generalizing virtual evidence (Subramanya and Bilmes, 2007; Pearl, 2014). Consequently, these diverse methods can be easily combined within a single framework for mutual amplification.

Third, we show that our problem formulation yields a well-defined learning objective (maximizing conditional likelihood of virtual evidence). We proposed a modular learning approach by decomposing the optimization over the supervision and prediction modules, using variational EM, which enables us to apply state-of-the-art methods for probabilistic logic and deep learning.

Finally, we applied DPL to biomedical machine reading (Quirk and Poon, 2017; Peng et al., 2017). Biomedicine offers a particularly attractive application domain for exploring indirect supervision. Biomedical literature grows by over one million each year[1], making it imperative to develop machine reading methods for automating knowledge curation (Figure 2). While crowd sourcing is hardly applicable, there are rich domain knowledge and structured resources to exploit for indirect supervision. Using cross-sentence relation extraction and entity linking as case studies, we show that distant supervision, data programming, and joint inference can be seamlessly combined in DPL to substantially improve machine reading accuracy, without requiring any manually labeled examples.[2]

2 Related Work

Distant supervision This paradigm was first introduced for binary relation extraction (Craven and Kumlien, 1999; Mintz et al., 2009). In its simplest form, distant supervision generates a positive example if an entity pair with a known relation co-occurs in a sentence, and samples negative examples from co-occurring entity pairs not known to have the given relation. It has recently been extended to cross-sentence relation extraction (Quirk and Poon, 2017; Peng et al., 2017). In principle, one simply looks beyond single sentences for co-occurring entity pairs. However, this can introduce many false positives and prior work used a small sliding window and filtering (minimal-span) to mitigate training noise. Even so, accuracy is relatively low. Both Quirk and Poon (2017) and Peng et al. (2017) used ontology-based string matching for entity linking, which also incurs many false positives, as biomedical entities are highly ambiguous (e.g., PDF and AAAS are gene names). Distant supervision for entity linking is relatively underexplored, and prior work generally focuses on Freebase entities, where links to the corresponding Wikipedia articles are available for learning (Huang et al., 2015).

Data Programming Instead of annotated examples, domain experts are asked to produce labeling functions, each of which assigns a label to an instance if the input satisfies certain conditions, often specified by simple rules (Ratner et al., 2016). This paradigm is useful for semantic tasks, as high-precision text-based rules are often easy to come by. However, there is no guarantee on broad coverage, and labeling functions are still noisy and may contradict with each other. The common denoising strategy assumes that labeling functions make random mistakes, and focuses on estimating their

[1] http://ncbi.nlm.nih.gov/pubmed

[2] The DPL code and datasets will be made available at http://hanover.azurewebsites.net.

accuracy and correlation (Ratner et al., 2016; Bach et al., 2017). A more sophisticated strategy also models instance-level labels and uses instance embedding to estimate instance-level weight for each labeling function (Liu et al., 2017).

Joint Inference Distant supervision and data programming focus on infusing weak supervision on individual labels. Additionally, there is rich linguistic and domain knowledge that does not specify values for individual labels, but imposes hard or soft constraints on their joint distribution. For example, if two mentions are coreferent, they should agree on entity properties (Poon and Domingos, 2008). There is a rich literature on joint inference for NLP applications. Notable methodologies include constraint-driven learning (Chang et al., 2007), general expectation (Druck et al., 2008), posterior regularization (Ganchev et al., 2010), and probabilistic logic (Poon and Domingos, 2008). Constraints can be imposed on relational instances or on model expectations. Learning and inference are often tailor-made for each approach, including beam search, primal-dual optimization, weighted satisfiability solvers, etc. Recently, joint inference has also been used in denoising distant supervision. Instead of labeling all co-occurrences of an entity pair with a known relation as positive examples, one only assumes that at least one instance is positive (Hoffmann et al., 2011; Lin et al., 2016).

Probabilistic Logic Probabilistic logic combines logic's expressive power with graphical model's capability in handling uncertainty. A representative example is Markov logic (Richardson and Domingos, 2006), which define a probability distribution using weighted first-order logical formulas as templates for a Markov model. Probabilistic logic has been applied to incorporating indirect supervision for various NLP tasks (Poon and Domingos, 2007, 2008; Poon and Vanderwende, 2010), but its expressive power comes at a price: learning and inference are generally intractable, and end-to-end modeling often requires heavy approximation (Kimmig et al., 2012). In DPL, we limit the use of probabilistic logic to modeling indirect supervision in the supervision module, leaving end-to-end modeling to deep neural network in the prediction module. This alleviates the computational challenges in probabilistic logic, while leveraging the strength of deep learning in distilling complex patterns from high-dimension data.

Knowledge-Rich Deep Learning Infusing knowledge in neural network training is a long-standing challenge in deep learning (Towell and Shavlik, 1994). Hu et al. (2016a,b) first used logical rules to help train a convolutional neural network for sentiment analysis. DPL draws inspiration from their approach, but is more general and theoretically well-founded. Hu et al. (2016a,b) focused on supervised learning and the logical rules were introduced to augment labeled examples via posterior regularization (Ganchev et al., 2010). DPL can incorporate both direct and indirect supervision, including posterior regularization and other forms of indirect supervision. Like DPL, Hu et al. (2016b) also refined uncertain weights of logical rules, but they did it in a heuristic way by appealing to symmetry with standard posterior regularization. We provide a novel problem formulation using generalized virtual evidence, which shows that their heuristics is a special case of variational EM and opens up opportunities for other optimization strategies.

Deep generative models also combine deep learning with probabilistic models, but focus on uncovering latent factors to support generative modeling and semi-supervised learning (Kingma and Welling, 2013; Kingma et al., 2014). Knowledge infusion is limited to introducing structures among the latent variables (e.g., Markov chain) (Johnson et al., 2016). In DPL, we focus on learning a discriminative model for predicting the latent labels, using a probabilistic model defined by probabilistic logic to inject indirect supervision.

3 Deep Probabilistic Logic

In this section, we introduce deep probabilistic logic (DPL) as a unifying framework for indirect supervision. Label decisions are modeled as latent variables. Indirect supervision is represented as generalized virtual evidence, and learning maximizes the conditional likelihood of virtual evidence given input. We first review the idea of virtual evidence and show how it can be generalized to represent any form of indirect supervision. We then formulate the learning objective and show how it can be optimized using variational EM.

Given a prediction task, let $\mathcal{X}$ denote the set of possible inputs and $\mathcal{Y}$ the set of possible outputs. The goal is to train a prediction module $\Psi(x, y)$ that scores output y given input x. Without loss of generality, we assume that $\Psi(x, y)$ defines the conditional probability $P(y|x)$ using a deep neural

network with a softmax layer at the top. Let $X = (X_1, \cdots, X_N)$ denote a sequence of inputs and $Y = (Y_1, \cdots, Y_N)$ the corresponding outputs. We consider the setting where Y are unobserved, and $\Psi(x, y)$ is learned using indirect supervision.

Virtual evidence Pearl (Pearl, 2014) first introduced the notion of virtual evidence, which has been used to incorporate label preference in semi-supervised learning (Reynolds and Bilmes, 2005; Subramanya and Bilmes, 2007; Li, 2009) and grounded learning (Parikh et al., 2015). Suppose we have a prior belief on the value of y, it can be represented by introducing a binary variable v as a dependent of y such that $P(v = 1|y = l)$ is proportional to the prior belief of $y = l$. $v = 1$ is thus an observed evidence that imposes soft constraints over y. Direct supervision (i.e., observed label) for y is a special case when the belief is concentrated on a specific value $y = l^*$ (i.e., $P(v = 1|y = l) = 0$ for any $l \neq l^*$). The virtual evidence v can be viewed as a reified variable for a potential function $\Phi(y) \propto P(v = 1|y)$. This enables us to generalize virtual evidence to arbitrary potential functions $\Phi(X, Y)$ over the inputs and outputs. In the rest of the paper, we will simply refer to the potential functions as virtual evidences, without introducing the reified variables explicitly.

DPL Let $K = (\Phi_1, \cdots, \Phi_V)$ be a set of virtual evidence derived from prior knowledge. DPL comprises of a supervision module over K and a prediction module over all input-output pairs (Figure 1), and defines a probability distribution:

$$P(K, Y|X) \propto \prod_v \Phi_v(X, Y) \cdot \prod_i \Psi(X_i, Y_i)$$

Without loss of generality, we assume that virtual evidences are log-linear factors, which can be compactly represented by weighted first-order logical formulas (Richardson and Domingos, 2006). Namely, $\Phi_v(X, Y) = \exp(w_v \cdot f_v(X, Y))$, where $f_v(X, Y)$ is a binary feature represented by a first-order logical formula. A hard constraint is the special case when $w_v = \infty$ (in practice, it suffices to set it to a large number, e.g., 10). In prior use of virtual evidence, w_v's are generally pre-determined from prior knowledge. However, this may be sub-optimal. Therefore, we consider a general Bayesian learning setting where each w_v is drawn from a pre-specified prior distribution $w_v \sim P(w_v|\alpha_v)$. Fixed w_v amounts to the special case when the prior is concentrated on the preset value. For uncertain w_v's, we can compute their maximum a posteriori (MAP) estimates and/or quantify the uncertainty.

Distant supervision Virtual evidence for distant supervision is similar to that for direct supervision. For example, for relation extraction, distant supervision from a knowledge base of known relations will set $f_{KB}(X_i, Y_i) = \mathbb{I}[\texttt{In-KB}(X_i, r) \wedge Y_i = r]$, where $\texttt{In-KB}(X_i, r)$ is true iff the entity tuple in X_i is known to have relation r in the KB.

Data programming Virtual evidence for data programming is similar to that for distant supervision: $f_L(X_i, Y_i) = \mathbb{I}[L(X_i) = Y_i]$, where $L(X_i)$ is a labeling function provided by domain experts. Labeling functions are usually high-precision rules, but errors are still common, and different functions may assign conflicting labels to an instance. Existing denoising strategy assumes that each function makes random errors independently, and resolves the conflicts by weighted votes (Ratner et al., 2016). In DPL, this can be done by simply treating error probabilities as uncertain parameters and inferring them during learning.

Joint inference Constraints on instances or model expectations can be imposed by introducing the corresponding virtual evidence (Ganchev et al., 2010) (Proposition 2.1). The weights can be set heuristically (Chang et al., 2007; Mann and McCallum, 2008; Poon and Domingos, 2008) or iteratively via primal-dual methods (Ganchev et al., 2010). In addition to instance-level constraints, DPL can incorporate arbitrary high-order soft and hard constraints that capture the interdependencies among multiple instances. For example, identical mentions in proximity probably refer to the same entity, which is useful for resolving ambiguous mentions by leveraging their unambiguous coreferences (e.g., an acronym in apposition of the full name). This can be represented by the virtual evidence $f_{\texttt{Joint}}(X_i, Y_i, X_j, Y_j) = \mathbb{I}[\texttt{Coref}(X_i, X_j) \wedge Y_i = Y_j]$, where $\texttt{Coref}(X_i, X_j)$ is true iff X_i and X_j are coreferences. Similarly, the common denoising strategy for distant supervision replaces the mention-level constraints with type-level constraints (Hoffmann et al., 2011). Suppose that $X_E \subset X$ contains all X_i's with co-occurring entity tuple E. The new constraints simply impose that, for each E with known relation $r \in KB$, $Y_i = r$ for at least one $X_i \in X_E$. This can be represented by a high-order factor on $(X_i, Y_i : X_i \in X_E)$.

Algorithm 1 DPL Learning

Input: Virtual evidences $K = \Phi_{1:V}$, deep neural network Ψ, inputs $X = (X_1, \cdots, X_N)$, unobserved outputs $Y = (Y_1, \cdots, Y_N)$.
Output: Learned prediction module Ψ^*
Initialize: $\Phi^0 \sim$ priors, $\Psi^0 \sim$ uniform.
for $t = 1 : T$ **do**

$$q^t(Y) \leftarrow \arg\min_q \; D_{KL}(\prod_i q_i(Y_i) \;||\; \prod_v \Phi_v^{t-1}(X,Y) \cdot \prod_i \Psi^{t-1}(X_i, Y_i))$$

$$\Phi^t \leftarrow \arg\min_\Phi \; D_{KL}(q^t(Y) \;||\; \prod_v \Phi_v(X,Y))$$

$$\Psi^t \leftarrow \arg\min_\Psi \; D_{KL}(q^t(Y) \;||\; \prod_i \Psi(X_i, Y_i))$$

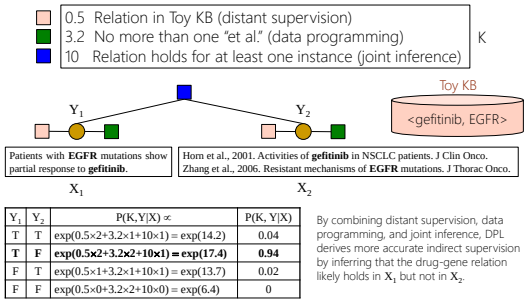
end for
return $\Psi^* = \Psi^T$.

Parameter learning Learning in DPL maximizes the conditional likelihood of virtual evidences $P(K|X)$. We can directly optimize this objective by summing out latent Y to compute the gradient and run backpropagation. In this paper, however, we opted for a modular approach using variational EM. See Algorithm 1.

In the E-step, we compute a variational approximation $q(Y) = \prod_i q_i(Y_i)$ by minimizing its KL divergence with $P(Y|K,X)$, which amounts to computing marginal probabilities $q_i(Y_i) = P(Y_i|K,X) = \sum_{Y_{-i}} P(Y_i, Y_{-i}|K,X)$, with current parameters Φ, Ψ. This is a standard probabilistic inference problem. Exact inference is generally intractable, but there are a plethora of approximate inference methods that can efficiently produce an estimate. We use loopy belief propagation (Murphy et al., 1999) in this paper, by conducting message passing in $P(K,Y|X)$ iteratively. Note that this inference problem is considerably simpler than end-to-end inference with probabilistic logic, since the bulk of the computation is encapsulated by Ψ.

Inference with high-order factors of large size can be challenging, but there is a rich body of literature for handling such structured factors in a principled way. In particular, in distant supervision denoising, we alter the message passing schedule so that each at-least-one factor will compute messages to its variables jointly by renormalizing their current marginal probabilities with noisy-or (Keith et al., 2017), which is essentially a soft version of dual decomposition (CarøE and Schultz, 1999).

In the M-step, we treat the variational approximation $q_i(Y_i)$ as probabilistic labels, and use them to optimize Φ and Ψ via standard supervised learning, which is equivalent to minimizing the KL



Y_1	Y_2	P(K,Y|X) ∝	P(K, Y|X)
T	T	exp(0.5×2+3.2×1+10×1) = exp(14.2)	0.04
T	**F**	**exp(0.5x2+3.2x2+10x1) = exp(17.4)**	**0.94**
F	T	exp(0.5×1+3.2×1+10×1) = exp(13.7)	0.02
F	F	exp(0.5×0+3.2×2+10×0) = exp(6.4)	0

By combining distant supervision, data programming, and joint inference, DPL derives more accurate indirect supervision by inferring that the drug-gene relation likely holds in X_1 but not in X_2.

Figure 3: Example of DPL combining various indirect supervision using probabilistic logic. The prediction module is omitted to avoid clutter.

divergence between the probabilistic labels and the conditional likelihood of Y given X under the supervision module (Φ) and prediction module (Ψ), respectively. For the prediction module, this optimization reduces to standard deep learning. Likewise, for the supervision module, this optimization reduces to standard parameter learning for log-linear models (i.e., learning all w_v's that are not fixed). Given the probabilistic labels, it is a convex optimization problem with a unique global optimum. Here, we simply use gradient descent, with the partial derivative for w_v being $\mathbb{E}_{\Phi(Y,X)}[f_v(X,Y)] - \mathbb{E}_{q(Y)}[f_v(X,Y)]$. For a tied weight, the partial derivative will sum over all features that originate from the same template. The second expectation can be done by simple counting. The first expectation, on the other hand, requires probabilistic inference in the graphical model. But it can be computed using belief propagation, similar to the E-step, except that the messages are limited to factors within the supervision module (i.e., messages from Ψ are not longer included). Convergence is usually fast, upon which the marginal for each Y_i is available, and $\mathbb{E}_{\Phi(Y,X)}[f_v(X,Y)]$ is simply the fraction of Y that renders $f_v(X,Y)$ to be true. Again, this parameter learning problem is much simpler than end-to-end learning with probabilistic logic, as it focuses on refining uncertain weights for indirect supervision, rather than learning complex input patterns for label prediction (handled in deep learning).

Example Figure 3 shows a toy example on how DPL combines various indirect supervision for predicting drug-gene interaction (e.g., gefitinib can be used to treat tumors with EGFR mutations). Indirect supervision is modeled by probabilistic logic, which defines a joint probability distribution over latent labeling decisions for drug-gene mention pairs in unlabeled text. Here, distant supervision

prefers classifying mention pairs of known relations, whereas the data programming formula opposes classifying instances resembling citations, and the joint inference formula ensures that at least one mention pair of a known relation is classified as positive. Formula weight signifies the confidence in the indirect supervision, and can be refined iteratively along with the prediction module.

Handling label imbalance One challenge for distant supervision is that negative examples are often much more numerous. A common strategy is to subsample negative examples to attain a balanced dataset. In preliminary experiments, we found that this was often suboptimal, as many informative negative examples were excluded from training. Instead, we restored the balance by up-weighting positive examples. In DPL, an additional challenge is that the labels are probabilistic and change over iterations. In this paper, we simply used hard EM, with binary labels set using 0.5 as the probability threshold, and the up-weighting coefficient recalculated after each E-step.

4 Biomedical Machine Reading

There is a long-standing interest in biomedical machine reading (e.g., Morgan et al. (2008); Kim et al. (2009)), but prior studies focused on supervised approaches. The advent of big biomedical data creates additional urgency for developing scalable approaches that can generalize to new reading tasks. For example, genome sequencing cost has been dropping faster than Moore's Law, yet oncologists can only evaluate tumor sequences for a tiny fraction of patients, due to the bottleneck in assimilating relevant knowledge from publications. Recently, Peng et al. (2017) formulated precision oncology machine reading as cross-sentence relation extraction (Figure 2) and developed the state-of-the-art system using distant supervision. While promising, their results still leave much room to improve. Moreover, they used heuristics to heavily filter entity candidates, with significant recall loss.

In this section, we use cross-sentence relation extraction as a case study for combining indirect supervision using deep probabilistic logic (DPL). First, we show that DPL can substantially improve machine reading accuracy in a head-to-head comparison with Peng et al. (2017), using the same entity linking method. Next, we apply DPL to entity linking itself and attain similar improvement. Finally, we consider further improving the recall by removing the entity filter. By applying DPL to joint entity linking and relation extraction, we more than doubled the recall in relation extraction while attaining comparable precision as Peng et al. (2017) with heavy entity filtering.

Evaluation Comparing indirect supervision methods is challenging as there is often no annotated test set for evaluating precision and recall. In such cases, we resort to the standard strategy used in prior work by reporting *sample precision* (estimated proportion of correct system extractions) and *absolute recall* (estimated number of correct system extractions). Absolute recall is proportional to recall and can be used to compare different systems (modulo estimation errors).

Datasets We used the same unlabeled text as Peng et al. (2017), which consists of about one million full text articles in PubMed Central (PMC)[3]. Tokenization, part-of-speech tagging, and syntactic parsing were conducted using SPLAT (Quirk et al., 2012), and Stanford dependencies (de Marneffe et al., 2006) were obtained using Stanford CoreNLP (Manning et al., 2014). For entity ontologies, we used DrugBank[4] and Human Gene Ontology (HUGO)[5]. DrugBank contains 8257 drugs; we used the subset of 599 cancer drugs. HUGO contains 37661 genes. For knowledge bases, we used the Gene Drug Knowledge Database (GDKD) (Dienstmann et al., 2015) and the Clinical Interpretations of Variants In Cancer (CIVIC)[6]. Together, they contain 231 drug-gene-mutation triples, with 76 drugs, 35 genes and 123 mutations.

4.1 Cross-sentence relation extraction

Let $e_1, \cdots, e_m$ be entity mentions in text T. Relation extraction can be formulated as classifying whether a relation R holds for $e_1, \cdots, e_m$ in T. To enable a head-to-head comparison, we used the same cross-sentence setting as Peng et al. (2017), where T spans up to three consecutive sentences and R represents the ternary interaction over drugs, genes, and mutations (whether the drug is relevant for treating tumors with the given gene mutation).

Entity linking In this subsection, we used the entity linker from Literome (Poon et al., 2014) to identify drug, gene, and mutation mentions, as in Peng et al. (2017). This entity linker first identifies candidate mentions by matching entity names

[3] www.ncbi.nlm.nih.gov/pmc
[4] www.drugbank.ca
[5] www.genenames.org
[6] civic.genome.wustl.edu

Distant Supervision: GDKD, CIVIC
Data Programming (Entity)
Mention matches entity name exactly.
Mention not a stop word.
Mention not following figure designation.
Mention's POS tags indicate it is a noun.
Data Programming (Relation)
Less than 30% of words are numbers in each sentence.
No more than three consecutive numbers.
No more than two "et al".
No more than three tokens start with uppercase.
No more than three special characters.
No more than three keywords indicative of table or figure.
Entity mentions do not overlap.
Joint Inference: Relation holds in at least one instance.

Table 1: DPL combines three indirect supervision strategies for cross-sentence relation extraction

or synonyms in domain ontologies, then applies heuristics to filter candidates. The heuristics are designed to enhance precision, at the expense of recall. For example, one heuristics would filter candidates of length less than four, which eliminates key cancer genes such as ER or AKT.

Prediction module We used the same graph LSTM as in Peng et al. (2017) to enable head-to-head comparison on indirect supervision strategies. Briefly, a graph LSTM generalizes a linear-chain LSTM by incorporating arbitrary long-ranged dependencies, such as syntactic dependencies, discourse relations, coreference, and connections between roots of adjacent sentences. A word might have precedents other than the prior word, and its LSTM unit is expanded to include a forget gate for each precedent. See Peng et al. (2017) for details.

Supervision module We used DPL to combine three indirect supervision strategies for cross-sentence relation extraction (Table 1). For distant supervision, we used GDKD and CIVIC as in Peng et al. (2017). For data programming, we introduced labeling functions that aim to correct entity and relation errors. Finally, we incorporated joint inference among all co-occurring instances of an entity tuple with the known relation by imposing the at-least-one constraint (i.e., the relation holds for at least one of the instances). For development, we sampled 250 positive extractions from DPL using only distant supervision (Peng et al., 2017) and excluded them from future training and evaluation.

Experiment results We compared DPL with the state-of-the-art system of Peng et al. (2017). We also conducted ablation study to evaluate the impact of indirect-supervision strategies. For a fair comparison, we used the same probability threshold in all cases (an instance is classified as positive if the normalized probability score is at least 0.5). For each system, sample precision was estimated by sampling 100 positive extractions and manually determining the proportion of correct extractions by an author knowledgeable about this domain. Absolute recall is estimated by multiplying sample precision with the number of positive extractions.

System	Prec.	Abs. Rec.	Unique
Peng 2017	0.64	6768	2738
DPL + EMB	**0.74**	**8478**	**4821**
DPL	0.73	7666	4144
− DS	0.29	7555	4912
− DP	0.67	4826	2629
− DP (ENTITY)	0.70	7638	4074
− JI	0.72	7418	4011

Table 2: Comparison of sample precision and absolute recall (all instances and unique entity tuples) in test extraction on PMC. DPL + EMB is our full system using PubMed-trained word embedding, whereas DPL uses the original Wikipedia-trained word embedding in Peng et al. (2017). Ablation: DS (distant supervision), DP (data programming), JI (joint inference).

Pred. Mod.	Prec.	Abs. Rec.	Unique
BiLSTM	0.60	6243	3427
Graph LSTM	0.73	7666	4144

Table 3: Comparison of sample precision and absolute recall (all instances and unique entity tuples) in test extraction on PMC. Both use same indirect supervision and Wikipedia-trained word embedding.

Table 2 shows the results. DPL substantially outperformed Peng et al. (2017), improving sample precision by ten absolute points and raising absolute recall by 25%. Combining disparate indirect supervision strategies is key to this performance gain, as evident from the ablation results. While distant supervision remained the most potent source of indirect supervision, data programming and joint inference each contributed significantly. Replacing out-of-domain (Wikipedia) word embedding with in-domain (PubMed) word embedding (Pyysalo et al., 2013) also led to a small gain.

Peng et al. (2017) only compared graph LSTM and linear-chain LSTM in automatic evaluation, where distant-supervision labels were treated as ground truth. They found significant but relatively small gains by graph LSTM. We conducted additional manual evaluation comparing the two in

Distant Supervision: HGNC
Data Programming
No verbs in POS tags.
Mention not a common word.
Mention contains more than two characters or one word.
More than 30% of characters are upper case.
Mention contains both upper and lower case characters.
Mention contains both character and digit.
Mention contains more than six characters.
Dependency label from mention to parent indicative of direct object.
Joint Inference
Identical mentions nearby probably refer to the same entity.
Appositive mentions probably refer to the same entity.
Nearby mentions that match synonyms of same entity probably refer to the given entity.

Table 4: DPL combines three indirect supervision strategies for entity linking.

DPL. Surprisingly, we found rather large performance difference, with graph LSTM outperforming linear-chain LSTM by 13 absolute points in precision and raising absolute recall by over 20% (Table 3). This suggests that Peng et al. (2017) might have underestimated the performance gain by graph LSTM using automatic evaluation.

4.2 Entity linking

Let m be a mention in text and e be an entity in an ontology. The goal of entity linking is to predict $\texttt{Link}(\texttt{m}, \texttt{e})$, which is true iff m refers to e, for every candidate mention-entity pair m, e. We focus on genes in this paper, as they are particularly noisy.

Prediction module We used BiLSTM with attention over the ten-word windows before and after a mention. The embedding layer is initialized by word2vec embedding trained on PubMed abstracts and full text (Pyysalo et al., 2013). The word embedding dimension was 200. We used 5 epochs for training, with Adam as the optimizer. We set learning rate to 0.001, and batch size to 64.

Supervision module As in relation extraction, we combined three indirect supervision strategies using DPL (Table 4). For distant supervision, we obtained all mention-gene candidates by matching PMC text against the HUGO lexicon. We then sampled a subset of 200,000 candidate instances as positive examples. We sampled a similar number of noun phrases as negative examples. For data programming, we introduced labeling functions that used mention characteristics (longer names are less ambiguous) or syntactic context (genes are more likely to be direct objects and nouns). For joint inference, we leverage linguistic phenomena related to coreference (identical, appositive, or synonymous mentions nearby are likely coreferent).

System	Acc.	F1	Prec.	Rec.
String Match	0.18	0.31	0.18	1.00
DS	0.64	0.71	0.62	0.83
DS + DP	0.66	0.71	0.62	0.83
DS + DP + JI	**0.70**	**0.76**	0.68	0.86

Table 5: Comparison of gene entity linking results on a balanced test set. The string-matching baseline has low precision. By combining indirect supervision strategies, DPL substantially improved precision while retaining reasonably high recall.

	F1	Precision	Recall
GNormPlus	0.78	0.74	0.81
DPL	0.74	0.68	0.80

Table 6: Comparison of gene entity linking results on BioCreative II test set. GNormPlus is the state-of-the-art system trained on thousands of labeled examples. DPL used only indirect supervision.

Experiment results For evaluation, we annotated a larger set of sample gene-mention candidates and then subsampled a balanced test set of 550 instances (half are true gene mentions, half not). These instances were excluded from training and development. Table 5 compares system performance on this test set. The string-matching baseline has a very low precision, as gene mentions are highly ambiguous, which explains why Peng et al. (2017) resorted to heavy filtering. By combining indirect supervision strategies, DPL improved precision by over 50 absolute points, while retaining a reasonably high recall (86%). All indirect supervision strategies contributed significantly, as the ablation tests show. We also evaluated DPL on BioCreative II, a shared task on gene entity linking (Morgan et al., 2008). We compared DPL with GNormPlus (Wei et al., 2015), the state-of-the-art supervised system trained on thousands of labeled examples in BioCreative II training set. Despite using zero manually labeled examples, DPL attained comparable F1 and recall (Table 6). The difference is mainly in precision, which indicates opportunities for more indirect supervision.

4.3 Joint entity and relation extraction

An important use case for machine reading is to improve knowledge curation efficiency by offering extraction results as candidates for curators to vet. The key to practical adoption is attaining high recall with reasonable precision (Peng et al., 2017). The entity filter used in Peng et al. (2017) is not ideal in this aspect, as it substantially reduced recall. In this

System	Prec	Abs. Rec.	Unique
Peng 2017	0.31	11481	5447
DPL (RE)	0.52	17891	8534
+ EL (TRN)	0.55	**21881**	**11047**
+ EL (TRN/TST)	**0.61**	20378	10291

Table 7: Comparison of sample precision and absolute recall (all instances and unique entity tuples) when all gene mention candidates are considered. Peng et al. (2017) used distant supervision only. RE: DPL relation extraction. EL: using DPL entity linking in RE training (TRN) and/or test (TST).

Gene	Drug	Mut.	Gene-Mut.	Relation
27%	4%	20%	45%	24%

Table 8: Error analysis for DPL relation extraction.

subsection, we consider replacing the entity filter by the DPL entity linker Table 7. Specifically, we added one labeling function to check if the entity linker returns a normalized probability score above $p_{\texttt{TRN}}$ for gene mentions, and filtered test instances if the gene mention score is lower than $p_{\texttt{TST}}$. We set $p_{\texttt{TRN}} = 0.6$ and $p_{\texttt{TST}} = 0.3$ from preliminary experiments. The labeling function discouraged learning from noisy mentions, and the test-time filter skips an instance if the gene is likely wrong. Not surprisingly, without entity filtering, Peng et al. (2017) suffered large precision loss. All DPL versions substantially improved accuracy, with significantly more gains using the DPL entity linker.

4.4 Discussion

Scalability DPL is efficient to train, taking around 3.5 hours for relation extraction and 2.5 hours for entity linking in our PubMed-scale experiments, with 25 CPU cores (for probabilistic logic) and one GPU (for LSTM). For relation extraction, the graphical model of probabilistic logic contains around 7,000 variables and 70,000 factors. At test time, it is just an LSTM, which predicted each instance in less than a second. In general, DPL learning scales linearly in the number of training instances. For distant supervision and data programming, DPL scales linearly in the number of known facts and labeling functions. As discussed in Section 3, joint inference with high-order factors is more challenging, but can be efficiently approximated. For inference in probabilistic logic, we found that loopy belief propagation worked reasonably well, converging after 2-4 iterations. Overall, we ran variational EM for three iterations, using ten epochs of deep learning in each M-step. We found

Janjigian YY, Groen HJ, Horn L, Smit EF, Fu Y, Wang F et al. (2011) Activity and tolerability of afatinib (BIBW 2992) and **cetuximab** in NSCLC patients with acquired resistance to erlotinib or gefitinib. J Clin Oncol 29 (suppl): abstr 7525 14. Fujita Y Suda K Kimura H Matsumoto K Arao T Nagai T Highly sensitive detection of **EGFR T790M** mutation using colony hybridization predicts favorable prognosis of patients with lung cancer harboring activating EGFR mutation J Thorac Oncol 2012

E19 deletion ALK Solid Signet ring cells Intracytoplasmic No - Crizotinib - AWDa e 12 F/66 Never Adrenal/B M1 (IV) E20 **R803W** ALK Solid No No No +d **Erlotinib** PD AWDa 0.7 EGFR, epidermal growth factor receptor; PFS, progression-free survival; M , male; PY, pack-year; R, resection; E , exon; **KRAS**, v-Ki-ras2.

Figure 4: Example of relation-extraction errors corrected by DPL with additional indirect supervision.

these worked well in preliminary experiments and used the same setting in all final experiments.

Accuracy To understand more about DPL's performance gain over distant supervision, we manually inspected some relation-extraction errors fixed by DPL after training with additional indirect supervision. Figure 4 shows two such examples. While some data programming functions were introduced to prevent errors stemming from citations or flattened tables, none were directly applicable to these examples. This shows that DPL can generalize beyond the original indirect supervision.

While the results are promising, there is still much to improve. Table 8 shows estimated precision errors for relation extraction by DPL. (Some instances have multiple errors.) Entity linking can incorporate more indirect supervision. Joint entity linking and relation extraction can be improved by feeding back extraction results to linking. Improvement is also sorely needed in classifying mutations and gene-mutation associations. The prediction module can also be improved, e.g., by adding attention to graph LSTM. DPL offers a flexible framework for exploring all these directions.

5 Conclusion

We introduce DPL as a unifying framework for indirect supervision, by composing probabilistic logic with deep learning. Experiments on biomedical machine reading show that this enables novel combination of disparate indirect supervision methodologies, resulting in substantial gain in accuracy. Future directions include: combining DPL with deep generative models; exploring alternative optimization strategies; applications to other domains.

6 Acknowledgements

We thank David McAllester, Chris Quirk, and Scott Yih for useful discussions, and the three anonymous reviewers for helpful comments.

References

Stephen H Bach, Bryan He, Alexander Ratner, and Christopher Ré. 2017. Learning the structure of generative models without labeled data. In *International Conference on Machine Learning*, pages 273–282.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Claus C CarøE and Rüdiger Schultz. 1999. Dual decomposition in stochastic integer programming. *Operations Research Letters*, 24(1-2):37–45.

Ming-Wei Chang, Lev Ratinov, and Dan Roth. 2007. Guiding semi-supervision with constraint-driven learning. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 280–287.

Kevin Clark and Christopher D Manning. 2016. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 643–653.

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*.

Rodrigo Dienstmann, In Sock Jang, Brian Bot, Stephen Friend, and Justin Guinney. 2015. Database of genomic biomarkers for cancer drugs and clinical targetability in solid tumors. *Cancer Discovery*, 5.

Gregory Druck, Gideon Mann, and Andrew McCallum. 2008. Learning from labeled features using generalized expectation criteria. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 595–602. ACM.

Kuzman Ganchev, Jennifer Gillenwater, Ben Taskar, et al. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(Jul):2001–2049.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*, volume 1. MIT press Cambridge.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

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 Forty-Ninth Annual Meeting of the Association for Computational Linguistics*.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016a. Harnessing deep neural networks with logic rules. In *Proceedings of the 2016 Conference on Association for Computational Linguistics*.

Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric Xing. 2016b. Deep neural networks with massive learned knowledge. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1670–1679.

Hongzhao Huang, Larry Heck, and Heng Ji. 2015. Leveraging deep neural networks and knowledge graphs for entity disambiguation. *arXiv preprint arXiv:1504.07678*.

Matthew Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. 2016. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in neural information processing systems*, pages 2946–2954.

Katherine Keith, Abram Handler, Michael Pinkham, Cara Magliozzi, Joshua McDuffie, and Brendan O'Connor. 2017. Identifying civilians killed by police with distantly supervised entity-event extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1547–1557.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of bionlp'09 shared task on event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, pages 1–9. Association for Computational Linguistics.

Angelika Kimmig, Stephen Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. 2012. A short introduction to probabilistic soft logic. In *Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications*, pages 1–4.

Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Xiao Li. 2009. On the use of virtual evidence in conditional random fields. In *Proceedings of the 2009*

Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3, pages 1289–1297. Association for Computational Linguistics.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 2124–2133.

Liyuan Liu, Xiang Ren, Qi Zhu, Shi Zhi, Huan Gui, Heng Ji, and Jiawei Han. 2017. Heterogeneous supervision for relation extraction: A representation learning approach. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 46–56.

Gideon S Mann and Andrew McCallum. 2008. Generalized expectation criteria for semi-supervised learning of conditional random fields. *Proceedings of ACL-08: HLT*, pages 870–878.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the Fifty-Second Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the Forty-Seventh Annual Meeting of the Association for Computational Linguistics and the Fourth International Joint Conference on Natural Language Processing*.

Alexander A Morgan, Zhiyong Lu, Xinglong Wang, Aaron M Cohen, Juliane Fluck, Patrick Ruch, Anna Divoli, Katrin Fundel, Robert Leaman, Jörg Hakenberg, et al. 2008. Overview of biocreative ii gene normalization. *Genome biology*, 9(2):S3.

Kevin P Murphy, Yair Weiss, and Michael I Jordan. 1999. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 467–475. Morgan Kaufmann Publishers Inc.

Ankur P Parikh, Hoifung Poon, and Kristina Toutanova. 2015. Grounded semantic parsing for complex knowledge extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 756–766.

Judea Pearl. 2014. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier.

Nanyun Peng, Hoifung Poon, Chris Quirk, and Kristina Toutanova Wen tau Yih. 2017. Cross-sentence n-ary relation extraction with graph lstms. *Transactions of the Association for Computational Linguistics*, 5:101–115.

Hoifung Poon and Pedro Domingos. 2007. Joint inference in information extraction. In *AAAI*, volume 7, pages 913–918.

Hoifung Poon and Pedro Domingos. 2008. Joint unsupervised coreference resolution with markov logic. In *Proceedings of the conference on empirical methods in natural language processing*, pages 650–659. Association for Computational Linguistics.

Hoifung Poon, Chris Quirk, Charlie DeZiel, and David Heckerman. 2014. Literome: PubMed-scale genomic knowledge base in the cloud. *Bioinformatics*, 30(19).

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

S. Pyysalo, F. Ginter, H. Moen, T. Salakoski, and S. Ananiadou. 2013. Distributional semantics resources for biomedical text processing. In *Proceedings of LBM 2013*, pages 39–44.

Chris Quirk, Pallavi Choudhury, Jianfeng Gao, Hisami Suzuki, Kristina Toutanova, Michael Gamon, Wentau Yih, and Lucy Vanderwende. 2012. MSR SPLAT, a language analysis toolkit. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics:Demonstration Session*.

Chris Quirk and Hoifung Poon. 2017. Distant supervision for relation extraction beyond the sentence boundary. In *Proceedings of the Fifteenth Conference on European chapter of the Association for Computational Linguistics*.

Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016. Data programming: Creating large training sets, quickly. In *Advances in Neural Information Processing Systems*, pages 3567–3575.

Sheila M Reynolds and Jeff A Bilmes. 2005. Part-of-speech tagging using virtual evidence and negative training. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 459–466. Association for Computational Linguistics.

Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. *Machine learning*, 62(1-2):107–136.

Amarnag Subramanya and Jeff Bilmes. 2007. Virtual evidence for training speech recognizers using partially labeled data. In *The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 165–168. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Geoffrey G Towell and Jude W Shavlik. 1994. Knowledge-based artificial neural networks. *Artificial intelligence*, 70(1-2):119–165.

Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. 2015. Gnormplus: an integrative approach for tagging genes, gene families, and protein domains. *BioMed research international*, 2015.