Using Commonsense Knowledge to Automatically Create (Noisy) Training Examples from Text

Sriraam Natarajan, Jose Picado, Tushar Khot*, Kristian Kersting+, Christopher Re*, Jude Shavlik*
Wake Forest University, USA
* University of Wisconsin-Madison, USA
+ Fraunhofer IAIS, University of Bonn, Germany
{nataras, picajm11}@wfu.edu, tushar@cs.wisc.edu, kristian.kersting@iais.fraunhofer.de, {chrisre, shavlik}@cs.wisc.edu

Abstract

One of the challenges to information extraction is the requirement of human annotated examples. Current successful approaches alleviate this problem by employing some form of distant supervision i.e., look into knowledge bases such as Freebase as a source of supervision to create more examples. While this is perfectly reasonable, most distant supervision methods rely on a hand coded background knowledge that explicitly looks for patterns in text. In this work, we take a different approach – we create weakly supervised examples for relations by using commonsense knowledge. The key innovation is that this commonsense knowledge is completely independent of the natural language text. This helps when learning the full model for information extraction as against simply learning the parameters of a known CRF or MLN. We demonstrate on two domains that this form of weak supervision yields superior results when learning structure compared to simply using the gold standard labels.

1 Introduction

Supervised learning is one of the popular approaches to information extraction from natural language text where the goal is to learn relationships between attributes of interest – learn the individuals employed by a particular organization, identifying the winners and losers in a game, etc. There have been two forms of supervised learning particularly used for this task: First, is the pure supervised learning approach. For instance, the NIST Automatic Content Extraction (ACE) RDC 2003 and 2004 corpora, has over 1000 documents that have human labeled relations leading to over 16000 relations being mentioned in the documents (**?**). ACE systems then use textual features – lexical, syntactic and semantic features – to learn the mentions of the target relations (**?**; **?**).

But pure supervised approaches are quite limited in scalability due to the requirement of high quality labels. An attractive very successful second approach is *distant supervision* where, labels of relations in the text are created by applying a heuristic in a common knowledge base such as Freebase (**?**; **?**; **?**). An important property of such methods is that the quality of the labels are crucially dependent on the heuristic used to map the relations to the knowledge base. Consequently, there have been several approaches that aim to improve the quality of these labels ranging from casting the problem as multi-instance learning (**?**; **?**) to using patterns that frequently appear in the text (**?**).

We take a different approach of creating more examples to the supervised learner based on *weak supervision* (**?**). We propose to use commonsense knowledge to create sets of entities that are "potential" relations. This commonsense knowledge is written by a domain expert in a probabilistic logic formalism called as *Markov Logic Networks* (MLN) (**?**). The algorithm then learns the parameters of these MLN clauses (we call them as *world MLN* – WMLN – to reflect that they are non-linguistic models) from a knowledge base such as Wikipedia. During the information extraction phase, unlabeled text are then parsed through some entity resolution parser to identify potential entities. Then these entities are provided as queries to the world MLN which uses data from non-NLP sources such as Wikipedia to then predict the posterior probability of relations between these entities. These predicted relations become the probabilistic (weakly supervised) examples for the next step.

Our hypothesis is – which we verify empirically – that the use of world knowledge will help in learning from natural language text. This is particularly true when there is a need to learn a model without any prior structure (a CRF or a MRF or a MLN) since the number of examples needed to learn the model can be very large. These weakly supervised examples can then augment the gold standard examples to improve the quality of the learned models. So far, the major hurdle to learning structure in information extraction is the number of features which can be very large leading to increased complexity in the search. We employ a recently successful probabilistic logic learning algorithm based on *Relational Functional Gradient Boosting* (RFGB) (**?**; **?**; **?**; **?**) for learning structure of these models.

Inspired by this success, we adapt the RFGB algorithm to learn in the presence of probabilistic examples by explicitly opitimizing the KL-divergence. We then employ RFGB in two different tasks, the first task is learning to jointly predict game winners and losers from NFL news articles[1]. We learned from 50 labeled documents and used 400 unlabeled documents. For the unlabeled documents, we used a common publicly available knowledge base such as Freebase to perform inference on the game winners and losers. We also

[1]LDC catalog number LDC2009E112

evaluate our algorithm on a second task, that of classifying documents as either *football* or *soccer* articles. We perform 5-fold cross validation on these tasks and our experiments conclusively prove that the proposed approach outperforms simply learning from gold standard data.

To summarize, the key contribution of the paper is that when we can bias the learner with examples created from commonsense knowledge, we can distantly learn structure. Our proposed algorithm has two distinct phases:

1. *Weakly supervised example generation* phase, where the goal is to use commonsense knowledge (WMLN). This WMLN could contain clauses such as "Higher ranked teams are more likely to win", "Home team are more likely to win", etc. Given this WMLN, parameters (weights) are learned from a knowledge base by looking at the previously completed games. Of course, these weights could also be provided by the domain expert. Once these weights are learned, predictions are made on entities extracted from unlabeled text and these predictions serve as weakly supervised examples for our next phase. Note that this phase is independent of the linguistic information and simply relies on world knowledge.
2. *Information extraction phase*, where the noisy examples are combined with some "gold standard" examples and a RDN is learned using textual features from the gold standard and the weakly supervised documents. Note that this phase only uses the text information for learning the model. The world knowledge is ignored when learning from linguistic features.

While our proposed approach has been presented in the context of information extraction, the idea of using outside world knowledge to create examples is more broadly applicable. For instance, this type advice can be used for labeling tasks (**?**) or to shape rewards in reinforcement learning (**?**) or to improve the number of examples in a medical task. Such advice can also be used to provide guidance to a learner in unforseen situations (**?**).

We proceed as follows: after reviewing the related work, we present the two phases of our approach in greater detail. We then present the experimental set up and results on the NFL task before concluding by pointing out future research directions.

2 Related work

2.1 Distant Supervision

As mentioned above, our approach is very similar to the *distant supervision* approaches (**?**; **?**) used to generate more training examples based on a knowledge base. These approaches use an external knowledge base to obtain a set of related entities. Sentences in which any of these related entities are mentioned, are now considered to be positive training examples. These examples along with the few annotated examples are provided to the learning algorithm. These approaches assume that the sentences that mention the related entities express the given relation. Riedel et al. (**?**) relax this assumption by introducing a latent variable for each mention pair to indicate whether the relation is mentioned or not. This work was further extended to allow overlapping relations between the same pair of entities (e.g. `Founded(Jobs, Apple)` and `CEO-of(Jobs, Apple)`) by modifying the latent variable to indicate the type of relation expressed by the sentence (**?**). In our approach, we define a model based on non-linguistic common sense knowledge to generate the distant supervision examples. Although we rely on a knowledge base to obtain the relevant features for our model, one can imagine tasks where such features are available as inputs or extracted further up in a pipeline.

2.2 Statistical Relational Learning

Most NLP approaches define a set of features relevant to the task and use propositional methods such as logistic regression. To obtain these features, they use structured output such as parse trees, dependency graphs, etc. obtained from a NLP toolkit. Recently, there has been a focus of employing Statistical Relational models that combine the expressiveness of first-order logic and the ability of probability theory to model uncertainty.

Many tasks such as BioNLP (**?**) and TempEval (**?**) involve multiple relations that need to be extracted jointly. Moreover, there are constraints on these relations, which are either defined by the task or by the user. To address these issues, Chambers and Jurafsky (**?**) defined the constraints using integer linear programming to jointly extract a consistent set of temporal relations. SRL models, on the other hand, can define the constraints much easily using first-order logic and can learn the model based on these constraints. As a result, SRL models, namely Markov Logic Networks (MLNs) (**?**), have been used for these tasks (**?**; **?**; **?**). But most of these approaches still relied on generating features from structured data. In our approach, we represent the structured data (e.g. parse trees) obtained from the Stanford toolkit using first-order logic and learn the structure of a SRL model called as Relational Dependency Networks (RDN) (**?**), to discover these features. Relational Dependency Networks (RDNs) are SRL models that consider a joint distribution as a product of conditional distributions. One of the important advantages of RDNs is that the models are allowed to be cyclic. As shown in the next section, we use MLNs to specify the weakly supervised world knowledge.

3 Structure Learning for Information Extraction Using Weak Supervision

One of the most important challenges facing many natural language tasks is the paucity of the "gold standard" examples. We outline our proposed method in detail in this section. Our method consists of two distinct phases: *weak supervision phase* where we create weakly supervised examples based on commonsense knowledge and *information extraction phase* where we learn the structure and parameters of the models that predict relations using textual features.

3.1 Weak Supervision Phase

We now explain our first phase in detail. As mentioned earlier, the key challenge in information extraction is obtaining annotated examples. To address this problem, we employ a method that is commonly taken by humans. For instance, consider reading a newspaper sports section about a particular sport (say NFL). Before we even read the article, we have an inherent *inductive bias* – we expect a high ranked team (particularly if it plays at home) to win. In other words, we rarely expect "upsets". We aim to formalize this notion by employing a model that captures this inductive bias to label examples in addition to the gold standard examples.

We employ the formalism of Markov Logic Networks (MLNs) to capture this world knowledge. MLNs (**?**) are relational undirected models where first-order logic formula correspond to the cliques of a Markov network and formula weights correspond to the clique potentials. A MLN can be instantiated as a Markov network with a node for each ground predicate (atom) and a clique for each ground formula. All groundings of the same formula are assigned the same weight, leading to the following joint probability distribution over all atoms: $P(X = x) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(x)\right)$, where $n_i(x)$ is the number of times the ith formula is satisfied by possible world x and Z is a normalization constant (as in Markov networks). Intuitively, a possible world where formula f_i is true one more time than a different possible world is e^{w_i} times as probable, all other things being equal. There have been several weight learning, structure learning and inference algorithms proposed for MLNs.

One of the reasons for using MLNs to capture commonsense knowledge is that MLNs provide an easy way for domain expert to specify the background knowledge as first-order logic clauses. Effective algorithms exist for learning the weights of these clauses given data. In our work, we use the Tuffy system (**?**) to learn the weights and perform inference on the MLNs. One of the key attractions of this Tuffy system is that it can scale to millions of documents and thus can provide a very efficient tool.

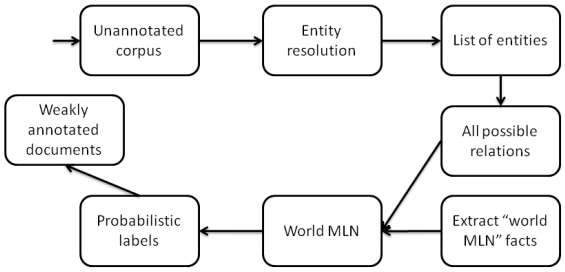

Figure 1: Steps involved in creation of weakly supervised examples.

Our proposed approach for weak supervision is presented in Figure **??**. Our first step is to employ a MLN that captures some commonsense knowledge about the domain of interest, called as *WMLN*. For the NFL domain, some of the rules that we used are shown in Table **??**. As can be observed from the table, our method uses some common knowledge such as "Home team is more likely to win the game" (first two clauses) and "High ranked team is more likely to win the game" (last two rules). Another clause that we found to be particularly useful is to say that "A team that is higher ranked and is the home team is more likely to win the game".

We learn the weights of these rules by extracting the previously played NFL games. Note that the rules are written without having the knowledge base in mind. These rules are simply written by the domain expert and they are softened using a knowledge base such as Wikipedia. The resulting weights are presented in the left column of the table. We used the games played in the last 20 years to compute these weights. Note that one could simply define a higher ranking using the following MLN clause where t denotes a team, r its rank, y the year of the ranking and hR the higher rank: $\infty \quad rank(t1, r1, y), rank(t2, r2, y), t1! = t2, r1 < r2 \rightarrow hR(t1, t2, y)$.

0.33	home(g, t) → winner(g, t)
0.33	away(g, t) → loser(g, t)
∞	exist t2 winner(g, t1), t1 != t2 → loser(g, t2)
∞	exist t2 loser(g, t1), t1 != t2 → winner(g, t2)
0.27	tInG(g, t1), tInG(g, t2), hR(t1, t2, y) → winner(g, t1)
0.27	tInG(g, t1), tInG(g, t2), hR(t1, t2, y) → loser(g, t2)

Table 1: A sample of WMLN clauses used for NFL task. t denotes a team, g denotes a game, y denotes the year, $tInG$ denotes that the team t plays in game g, $hR(t1, t2, y)$ denotes that $t1$ is ranked higher than $t2$ in year y.

Once the WMLN weights are learned, we proceed to create weakly supervised learning examples. To this effect, we identify interesting (unannotated) documents – for example, sport articles from different news web sites. We use a standard NLP tool such as the *Stanford NLP* toolkit to perform entity resolution to identify the potential teams, games and the year in the document. Once these entities are identified, we query the WMLN for obtaining the posterior on the relations between these entities – for example, game winner and loser relations from NFL articles. Recall that to perform inference, evidence is required. Hence, we use the games that have been potentially played between the two teams (again from previously played games that year) to identify the home, away and ranking of the teams. We used the rankings at the start of the year of the game as a pseudo reflection of the relative rankings between the teams.

The result of the inference process are the posterior probabilities of the relations between the entities extracted in the documents. The resulting relations are then used as annotations. One simple annotation scheme is using the MAP estimate (i.e., if the probability of a team being a winner is greater than the probability of being the loser, the relation becomes positive example for winner and a negative example for loser). An alternative method would be to use a method that directly learns from probabilistic labels which we focus in this work by modifying the learning algorithm. Choosing the MAP would make a strong commitment about several examples on the borderline. Note that since our world knowledge is independent of the text, it may be the

case that in some examples perfect labeling is not possible. In such cases, using a softer labeling method would be more beneficial. Hence, it is necessary to learn from noisy labels which we do by adapting the existing algorithm. Now the examples are ready for our next step – learning the model for *information extraction*.

3.2 Learning for Information Extraction

Once the weakly supervised examples are created, the next step is inducing the relations. In order to do so, we employ the procedure presented in Figure **??**. We run both the gold standard and weakly supervised annotated documents through Stanford NLP toolkit to create relational linguistic features – lexical, syntactic and semantic features. Once these features are created, we run the boosted RDN learner by Natarajan et al. (**?**). This allows us to create a joint model between the target relations, for example, game winner and losers. We now briefly describe the adaptation of boosted RDN to this task.

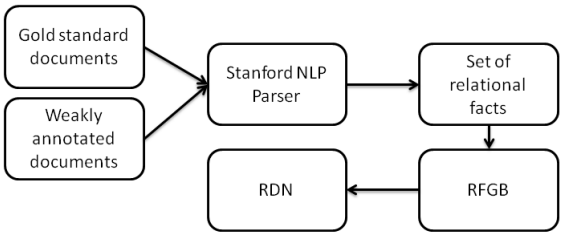

Figure 2: Steps involved in learning using probabilistic examples.

Assume that the training examples are of the form $(\mathbf{x}_i, y_i)$ for $i = 1, ..., N$ and $y_i \in \{1, ..., K\}$. We use $\mathbf{x}$ to denote the vector of features which in our case are lexical features and ys correspond to target (game winners and loser) relations. Relational models tend to consider training instances as "mega examples" where each example represents all instances of a particular group (example, an university, a research group etc). In our work, we consider each document to be a mega example and we do not learn across mega examples i.e., we do not consider cross document learning.

Given the above definitions, the goal is to fit a model $P(y|\mathbf{x}) \propto e^{\psi(y,\mathbf{x})}$ for every target relation y. Functional gradient ascent starts with an initial potential ψ_0 and iteratively adds gradients Δ_i. After m iterations, the potential is given by $\psi_m = \psi_0 + \Delta_1 + ... + \Delta_m$. Here, Δ_m is the functional gradient at episode m and is $\Delta_m = \eta_m \times E_{x,y}[\partial/\partial\psi_{m-1} log\, P(y|x; \psi_{m-1})]$, where η_m is the learning rate. Dietterich *et al.* (**?**) suggested evaluating the gradient at every position in every training example and fitting a regression tree to these derived examples i.e., fit a regression tree h_m on the training examples $[(x_i, y_i), \Delta_m(y_i; x_i)]$.

In our formalism, y corresponds to the target relations, for example $gameWinner$ and $gameLoser$ relation between a team and game mentioned in a sentence. $\mathbf{x}$ corresponds to all the relational facts associated with these mentions. To learn the model for a relation, say $gameWinner$, we start with an initial model ψ_0 which returns a constant regression value for all examples. Based on this initial model, we calculate the gradients for each example as the difference between the true label and current predicted probability. We learn a relational regression tree to fit the regression examples and add it to the current model. We now compute the gradients based on the updated model and repeat the process. Hence, in every subsequent iteration, we *fix* the errors made by the model. For further details about relational functional gradient boosting, we refer the readers to Natarajan et al. (**?**).

Since we use a probabilistic model to generate the weakly supervised examples, our training input examples will have probabilities associated with them based on the predictions from WMLN. We extend the relational functional gradient boosting approach to handle probabilistic examples by defining the loss function as the KL-divergence between the observed probabilities (shown using P_D) and predicted probabilities (shown using P). The functional gradients for the KL-divergence loss function can be shown to be the difference between the observed and predicted probabilities.

$$\begin{aligned}\Delta_m(x) &= \frac{\partial}{\partial\psi_{m-1}} \sum_{\hat{y}} P_D(y = \hat{y}) \log \left(\frac{P_D(y = \hat{y})}{P(y = \hat{y}|\psi_{m-1})} \right) \\ &= P_D(y = 1) - P(y = 1|\psi_{m-1})\end{aligned}$$

Hence the key idea in our work is to use probabilistic examples that we obtain from the weakly supervised phase as input to our structure learning phase along with gold standard examples (with $p = 1$ for positive examples), and their associated documents. Then a RDN is induced by learning to predict the different target relations jointly, using linguistic features created by the Stanford NLP toolkit. Since we are learning a RDN, we do not have to explicitly check for acyclicity. We chose to employ RDNs as they have been demonstrated to have the state-of-the-art performance in many tasks(**?**). We used the modified ordered gibbs sampler for inference.

4 Experimental results

In this section, we present the results of empirically validating our proposed approach on a natural language domain of predicting winners and losers. We compared the use of augmenting with weakly supervised examples against simply using the gold standard examples. Since we are also learning the structure of the model, we do not compare to other distant supervision methods directly but instead point out the state-of-the-art results in the problem.

4.1 Relation Extraction

The first data set that we evaluate our method is the National Football League (NFL) data set from LDC corpora[2]. This data set consists of articles of NFL games over the past two decades. This is essentially a natural language processing (NLP) task. The idea is to read the texts and identify concepts such as *winner*, and *loser* in the text. As an easy example, consider the text, "Packers defeated Cowboys $28-14$ in Saturday's Superbowl game". Then, the goal is to identify

[2]http://www.ldc.upenn.edu

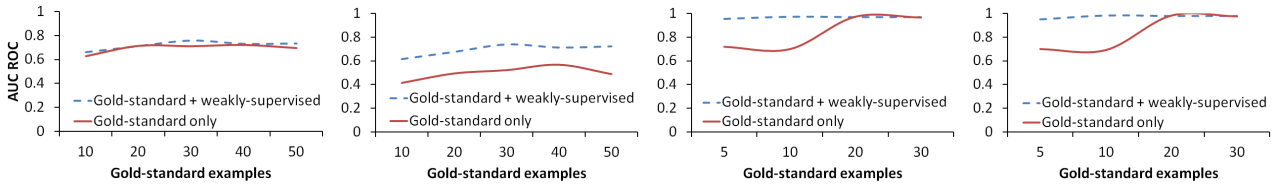

Figure 3: Results of predicting winners and losers: **(a)** AUC ROC. **(b)** AUC PR. Document classification: **(c)** AUC ROC. **(d)** AUC PR.

Greenbay Packers and *Dallas Cowboys* as the winner and loser respectively.

The corpus consists of articles, some of which are annotated with the target concepts. We consider only articles that have annotations of positive examples. There were 66 annotations of the relations. We used 16 of these annotations as the test set and performed training on (a subset) rest of the documents. In addition to the gold standard examples, we used articles from the NFL website[3] for weak supervision. In our experiment, we wanted to evaluate the impact of the weakly supervised examples. We used 400 weakly supervised examples. We varied the number of gold standard examples while keeping the number of weakly supervised examples constant. In another setting, we used no weakly supervised examples and simply varied the number of gold standard examples. The results were averaged over 5 runs of random selection of gold standard examples.

We measured the area under curves for both ROC and PR curves. Simply measuring the accuracy on the test set will not suffice in most structured problems, since predicting a majority class can lead in high performance. Hence we present AUC. The results are presented in Figure **??** where the performance measure is presented by varying the number of gold standard examples. As can be seen, in both metrics, the weakly supervised examples improve upon the usage of gold standard examples. The use of weakly supervised examples allows a jump start, a steeper learning curve and in the case of PR, a better convergence. It should be mentioned that while plotting every point, the set of the gold standard examples is kept constant for every run and the only difference is whether there are any weakly supervised examples added. For example, when plotting the results of 10 examples, for every run, the set of gold standard examples is the same. For the blue dashed curve, we add 400 more weakly supervised examples and this is repeated for 5 runs in which the 10 gold examples are drawn randomly. We also performed t-tests on all the points of the PR and ROC curves. For the PR curves, the use of weakly supervised learning yields statistically superior performance over the gold standard examples for all the points on the curves (with p-value < 0.05). For the ROC curves, significance occurs when using 10, and 30 examples. Since PR curves are more conservative than ROC curves, it is clear that the use of these weakly supervised examples improves the performance of the structure learner significantly. To understand whether weak supervision clearly helps, we performed another experiment using a baseline where we randomly assigned labels to the 400 examples. When combined with 50 gold standard examples, the performance decreased dramatically with AUC values of 0.58 for both ROC and PR curves which clearly shows that the weakly supervised labels help when learning the structure.

[3]http://www.nfl.com

4.2 Document Classification

To understand the general applicability of the proposed framework, we created another data set for evaluation. In this data set, the goal is to classify documents either as being *football(American)* or *soccer* articles. Hence the relation in this case is on the article (i.e.,*gametype(article,type)*). In order to do this, we extracted 30 football articles from the NFL website[4] and 30 soccer articles from the English Premier League (EPL) website[5] and annotated them manually as being football and soccer respectively. We used only the first paragraph of the articles for learning the models since it appeared that enough information is present in the first paragraph for learning an useful model. In addition, we used 45 articles for weak supervision. We used rules such as, "NFL teams play football", "EPL teams play soccer", "If the scores of both teams are greater than 10, then it is a football game", "If the scores of both teams are 0, then it is a soccer game".

All the rules mentioned above are essentially considered as "soft" rules. The weights of these rules were simply set to 100, 10, 1 to reflect the log-odds. Note that we could learn these weights as in the NFL cases, but the rules in this task are relatively simple and hence we simply set the weights manually. During the weak supervision phase, we used the entities mentioned in the documents as queries to the world MLN to predict the type of game that the entities correspond to. These predictions (probabilities) become the weak supervision for the learning phase. We labeled the 45 articles accordingly and combined them with the manually annotated articles.

As with the NFL data, we measured the AUC ROC and PR values by varying the number of gold standard examples. Again, in each run, to maintain consistency, we held the gold standard examples to be constant and simply added the weakly supervised examples. The results are presented in Figure **??**. The resulting figures show that as with the earlier case, weak supervision helps in improving the performance of the learning algorithm. We get a jump start and a steeper learning curve in this case as well. Again, the results

[4]http://www.nfl.com

[5]http://www.premierleague.com

are statistically significant for small number of gold standard examples. Both experiments conclusively prove that adding probabilistic examples as weak supervision enables our learning algorithm to improve upon its performance in the presence of small number of gold standard data thus validating the hypothesis that world knowledge helps when manual annotations are expensive.

5 Conclusion

One of the key challenges for applying learning methods in many real-world problems is the paucity of good quality labeled examples. While semi-supervised learning methods have been developed, we explore another alternative method of weak supervision – where the goal is to create examples of reasonable quality that can be relied upon. We considered the NLP tasks of relation extraction and document extraction to demonstrate the usefulness of the weak supervision. Our key insight is that weak supervision can be provided by a "domain" expert instead of a "NLP" expert and thus the knowledge is independent of the underlying problem but is close to the average human thought process – for example, sports fans. We used the weighted logic representation of Markov Logic networks to model the expert knowledge, learn the weights based on history and make predictions on the unannotated articles. We adapted the functional gradient boosting algorithm to learn relational dependency networks for prediciting the target relations. Our results demonstrate that our method significantly improves the performance thus reducing the need for gold standard examples.

Our proposed method is closely related to distant supervision methods. So it will be a very interesting future direction to combine the distant and weak supervision examples for structure learning. Combining weak supervision with other advice taking methods is another interesting direction. This method can be seen as giving advice about the examples, but AI has a long history of using advice on the model, the search space and examples. Hence, combining them might lead to a strong knowledge based system where the knowledge can be provided by a domain expert and not a AI/NLP expert. Finally, it is important to evaluate the proposed model in other similar tasks.

Acknowledgements Sriraam Natarajan, Tushar Khot and Jude Shavlik gratefully acknowledge the support of Defense Advanced Research Projects Agency (DARPA) Machine Reading Program and DEFT Program under Air Force Research Laboratory (AFRL) prime contract FA8750-09-C-0181 and FA8750-13-2-0039. Any opinions, findings, and conclusion expressed in this material are those of the authors and do not necessarily reflect the view of the DARPA, AFRL, or the US government.

References

Chambers, N., and Jurafsky, D. 2008. Jointly combining implicit constraints improves temporal ordering. In *EMNLP*.

Craven, M., and Kumlien, J. 1999. Constructing biological knowledge bases by extracting information from text sources.

Devlin, S.; Kudenko, D.; and Grzes, M. 2011. An empirical study of potential-based reward shaping and advice in complex, multi-agent systems. *Advances in Complex Systems* 14(2):251–278.

Dietterich, T.; Ashenfelter, A.; and Bulatov, Y. 2004. Training conditional random fields via gradient tree boosting. In *ICML*.

Domingos, P., and Lowd, D. 2009. *Markov Logic: An Interface Layer for AI*. San Rafael, CA: Morgan & Claypool.

Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; and Weld, D. S. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *HLT*.

Kersting, K., and Driessens, K. 2008. Non–parametric policy gradients: A unified treatment of propositional and relational domains. In *ICML*.

Khot, T.; Natarajan, S.; Kersting, K.; and Shavlik, J. 2011. Learning markov logic networks via functional gradient boosting. In *ICDM*.

Kim, J.; Ohta, T.; Pyysalo, S.; Kano, Y.; and Tsujii, J. 2009. Overview of BioNLP'09 shared task on event extraction. In *BioNLP Workshop Companion Volume for Shared Task*.

Kuhlmann, G.; Stone, P.; Mooney, R. J.; and Shavlik, J. W. 2004. Guiding a reinforcement learner with natural language advice: Initial results in robocup soccer. In *AAAI Workshop on Supervisory Control of Learning and Adaptive Systems*.

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009. Distant supervision for relation extraction without labeled data. In *ACL and AFNLP*.

Natarajan, S.; Joshi, S.; Tadepalli, P.; Kristian, K.; and Shavlik, J. 2011. Imitation learning in relational domains: A functional-gradient boosting approach. In *IJCAI*.

Natarajan, S.; Khot, T.; Kersting, K.; Guttmann, B.; and Shavlik, J. 2012. Gradient-based boosting for statistical relational learning: The relational dependency network case. *Machine Learning*.

Neville, J., and Jensen, D. 2007. Relational dependency networks. In Getoor, L., and Taskar, B., eds., *Introduction to Statistical Relational Learning*. 653–692.

Niu, F.; Ré, C.; Doan, A.; and Shavlik, J. W. 2011. Tuffy: Scaling up statistical inference in markov logic networks using an rdbms. *PVLDB* 4(6):373–384.

Poon, H., and Vanderwende, L. 2010. Joint inference for knowledge extraction from biomedical literature. In *NAACL*.

Riedel, S.; Chun, H.; Takagi, T.; and Tsujii, J. 2009. A markov logic approach to bio-molecular event extraction. In *BioNLP*.

Riedel, S.; Yao, L.; and McCallum, A. 2010. Modeling relations and their mentions without labeled text. In *ECML PKDD*.

Surdeanu, M., and Ciaramita, M. 2007. Robust information extraction with perceptrons. In *NIST ACE*.

Takamatsu, S.; Sato, I.; and Nakagawa, H. 2012. Reducing wrong labels in distant supervision for relation extraction. In *ACL*.

Torrey, L.; Shavlik, J.; Walker, T.; and Maclin, R. 2010. Transfer learning via advice taking. In *Advances in Machine Learning I*.

Verhagen, M.; Gaizauskas, R.; Schilder, F.; Hepple, M.; Katz, G.; and Pustejovsky, J. 2007. SemEval-2007 task 15: TempEval temporal relation identification. In *SemEval*.

Yoshikawa, K.; Riedel, S.; Asahara, M.; and Matsumoto, Y. 2009. Jointly identifying temporal relations with markov logic. In *ACL and AFNLP*.

Zhou, G.; Su, J.; Zhang, J.; and Zhang, M. 2005. Exploring various knowledge in relation extraction. In *ACL*.