Learning Relational Dependency Networks for Relation Extraction

Ameet Soni[1], Dileep Viswanathan[2], Jude Shavlik[3], and Sriraam Natarajan[2]

[1] Swarthmore College, Swarthmore, Pennsylvania, USA
`soni@cs.swarthmore.edu`
[2] Indiana University, Bloomington, Indiana, USA
`{diviswan, natarasr}@indiana.edu`
[3] University of Wisconsin, Madison, Wisconsin, USA
`shavlik@cs.wisc.edu`

Abstract. We consider the task of KBP slot filling – extracting relation information from newswire documents for knowledge base construction. We present our pipeline, which employs Relational Dependency Networks (RDNs) to learn linguistic patterns for relation extraction. Additionally, we demonstrate how several components such as weak supervision, `word2vec` features, joint learning and the use of human advice, can be incorporated in this relational framework. We evaluate the different components in the benchmark KBP 2015 task and show that RDNs effectively model a diverse set of features and perform competitively with current state-of-the-art relation extraction methods.

1 Introduction

The problem of knowledge base population (KBP) – constructing a knowledge base (KB) of facts gleaned from a large corpus of unstructured data – poses several challenges for the NLP community. Commonly, this relation extraction task is decomposed into two subtasks – entity linking, in which entities are linked to already identified identities within the document or to entities in the existing KB, and slot filling, which identifies certain attributes about a target entity.

We present our system for KBP slot filling based on probabilistic logic formalisms and present the different components of the system. Specifically, we employ Relational Dependency Networks [14], a formalism that has been successfully used for joint learning and inference from stochastic, noisy, relational data. We consider our RDN system against the current state-of-the-art for KBP to demonstrate the effectiveness of our probabilistic relational framework. Additionally, we show how RDNs can effectively incorporate many popular approaches in relation extraction such as joint learning, weak supervision, `word2vec` features, and human advice, among others.

We provide a comprehensive comparison of various settings such as joint learning vs learning of individual relations, use of weak supervision vs gold standard labels, using expert advice vs only learning from data, etc. These questions are extremely interesting from a general machine learning perspective, but also

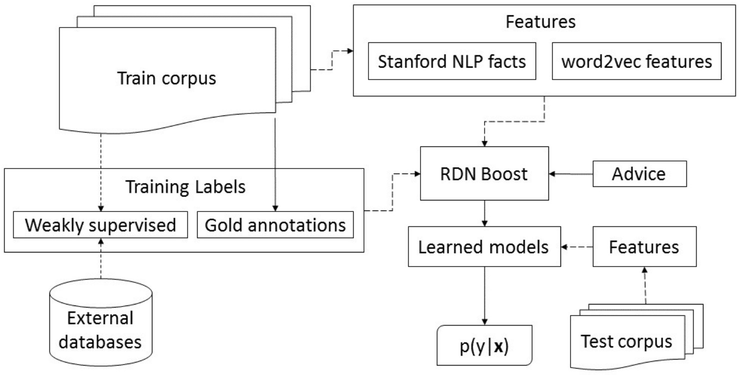

Fig. 1. Pipeline Full RDN relation extraction pipeline

critical to the NLP community. As we show empirically, the key contributions of this paper are as follows:

- Our RDN framework is competitive, and often superior, to state-of-the-art systems for KBP slot filling.
- RDNs successfully incorporate various types of features, including advice, joint learning, and `word2vec` features.
- Ours is the first KBP system to leverage *knowledge-based weak supervision* – a logic-based framework that we have previously shown to be complementary and often superior to distant supervision.

Some of the results such as human advice being useful in many relations and joint learning being beneficial in the cases where the relations are correlated among themselves are on the expected lines. However, some surprising observations include the fact that weak supervision and `word2vec` features are not as useful as expected, although further investigation is warranted.

We first present the proposed pipeline with all the different components of the learning system. Next we present the set of 14 relations that we learn on before presenting the experimental results. We finally discuss the results of these comparisons before concluding by presenting directions for future research.

2 Background

As a part of the Text Analysis Conference (TAC), NIST has supported several tasks related to Knowledge Base Population (KBP) including English Slot Filling [23]. The goal of this task is to mine a corpus of text data (e.g., newswire articles) for information on two specific categories of entities – persons and organizations. The type of information is predefined as relations (e.g., $parent(a, b)$ specifies that person b is a parent of person a).

Over the last several years, many approaches have been proposed across a spectrum of machine learning approaches. A common thread to these approaches

Table 1. Standard NLP Features Features derived from the training corpus used by our learning system. POS - part of speech. NE - Named Entity. DPR - root of dependency path tree.

Feature	**Description**
wordString	word with word id
wordPosition	location of the word
caselessWordString	word string in lower case
wordLemma	canonical form of word
isNEWord	whether word is NE
nextWords	two succeeding words
prevWords	two preceding words
nextPOS	POS for the succeeding words
prevPOS	POS for the preceding words
nextLemmas	canonical form of successors
prevLemmas	canonical form of predecessors
nextNE	succeeding NE phrases
prevNE	preceding NE phrases
lemmaBetween	canonical form of word occurring between two NEs
neBetween	word b/w two NEs is an NE
posBetween	POS of word b/w two NEs
Dependency Path	
rootChildLemma	canonical form of child of DPR
rootChildNER	child of DPR is NE
rootChildPOS	POS of child of DPR
rootLemma	lemma of DPR
rootNER	DPR is NER
rootPOS	POS of DPR

is the use of distant supervision [11]. The 2014 winner, DeepDive [17], leveraged Markov Logic Networks [2] with distant supervision to perform slot filling. RelationFactory [20] also utilizes distant supervision to train a highly modular pipeline that focuses on scaling and efficiency, employing several shallow classifiers (e.g., manually created patterns, learned rules, SVMs) for various tasks. Other approaches include multiple instance learning [10] and stacked ensembles which combine multiple submissions into a single framework [25].

3 Proposed Pipeline

We present the different aspects of our pipeline, depicted in Figure 1.

3.1 Feature Generation

Given a training corpus of raw text documents, our learning algorithm first converts these documents into a set of facts (i.e., features) that are encoded in first order logic (FOL). Raw text is processed using the Stanford CoreNLP Toolkit[4] [6] to extract parts-of-speech, word lemmas, etc. as well as generate

[4] `http://stanfordnlp.github.io/CoreNLP/`

Table 2. Rules for KB Weak Supervision A sample of knowledge-based rules for weak supervision. The first value defines a weight, or confidence in the accuracy of the rule. The target relation appears at the end of each clause. "PER", "ORG", "NUM" represent entities that are persons, organizations, and numbers, respectively.

Weight	MLN Clause
1.0	entityType(a, "PER"), entityType(b, "NUM"), nextWord(a, c), word(c, ","), nextWord(c, b) → age(a, b)
0.6	entityType(a, "PER"), entityType(b, "NUM"), prevLemma(b, "age") → age(a, b)
0.8	entityType(a, "PER"), entityType(b, "PER"), nextLemma(a, "mother") → parents(a, b)
0.8	entityType(a, "PER"), entityType(b, "PER"), nextLemma(a, "father") → parents(a, b)

parse trees, dependency graphs and named-entity recognition information. The full set of extracted features is listed in Table 1. These are then converted into features in prolog format and are given as input to the system.

In addition to the structured features from the output of Stanford toolkit, we also use deeper features based on `word2vec` [8] as input to our learning system. Standard NLP features tend to treat words as individual objects, ignoring links between words that occur with similar meanings or, importantly, similar contexts (e.g., city-country pairs such as *Paris – France* and *Rome – Italy* occur in similar contexts). `word2vec` provide a continuous-space vector embedding of words that, in practice, capture many of these relationships [8, 9]. We use word vectors from Stanford[5] and Google[6].

We generated features from word vectors by finding words with high similarity in the embedded space. That is, we used word vectors by considering relations of the following form: $isCosSimilar(wordA, relationB, threshold)$, if a word has a high cosine similarity to any keyword (e.g., "father") for a particular relation (e.g., *parent*). Details can be found in Section 4.3. At a high level, these types of features would allow our learner to generate rules that connect unique words that occur in similar contexts (e.g., "husband" and "wife"). Standard features,instead, would require the same rule to be learned multiple times (e.g., once each for "husband", "wife", "partner", etc. as in Figure 2).

3.2 Weak Supervision

One difficulty with the KBP task is that very few documents come labeled with *gold standard labels*, and human annotation is prohibitively expensive beyond a few hundred documents. This is problematic for discriminative learning algorithms which excel when given a large supervised training corpus. To overcome this obstacle, we employ *weak supervision* – the use of external knowledge (e.g., a

[5] `http://nlp.stanford.edu/projects/glove/`
[6] `https://code.google.com/p/word2vec/`

database) to heuristically label examples. Following our work in Soni et al. [21], we employ our novel knowledge-based weak supervision approach, as opposed to the more traditional distant supervision which references an external database of known relations.

Knowledge-based weak supervision is based on previous work [13, 21] with the following insight: labels are typically created by “domain experts” who annotate the labels carefully, and who typically employ some inherent rules in their mind to create examples. For example, when identifying family relationship, we may have an *inductive bias* towards believing two persons in a sentence with the same last name are related, or that the words “son” or “daughter” are strong indicators of a parent relation. We call this *world knowledge* as it describes the domain (or the world) of the target relation.

For the KBP task, some rules that we used are shown in Table 2. For example, the first rule identifies any number following a person’s name and separated by a comma is likely to be the person’s age (e.g., “Sharon, 42”). The third and fourth rule provide examples of rules that utilize more textual features; these rules state the appearance of the lemma “mother” or “father” between two persons is indicative of a parent relationship. Previous results show this approach produces more examples with less overhead than distant supervision and can be employed where relevant database are not available.

To this effect, we encode the domain expert’s knowledge in the form of first-order logic rules with accompanying weights to indicate the expert’s confidence. We use the probabilistic logic formalism *Markov Logic Networks* [2] to perform inference on unlabeled text (e.g., the TAC KBP corpus). Potential entity pairs from the corpus are queried to the MLN, yielding (weakly-supervised) positive examples. We choose MLNs as they permit domain experts to easily write rules while providing a probabilistic framework that can handle noise, uncertainty, and preferences. We use the Tuffy system [16] to perform inference as it is robust and scales well to millions of documents[7].

3.3 Learning Relational Dependency Networks

Previous research [7] has demonstrated that joint inferences of the relations are more effective than considering each relation individually. Consequently, we have considered a formalism that has been successfully used for joint learning and inference from stochastic, noisy, relational data called Relational Dependency Networks (RDNs) [14, 12]. RDNs extend dependency networks (DN) [4] to the relational setting. The key idea in a DN is to approximate the joint distribution over a set of random variables as a product of their marginal distributions, i.e., $P(y_1, ..., y_n|\mathbf{X}) \approx \prod_i P(y_i|\mathbf{X})$. It has been shown that employing Gibbs sampling in the presence of a large amount of data allows this approximation to be particularly effective. Note that, one does not have to explicitly check for

[7] As the structure and weights are predefined by the expert, learning is not needed for our MLN

Table 3. Advice Rules Sample advice rules used for relation extraction. We employed a total of 72 such rules for our 14 relations.

Advice Rules
Entity preceded by a number and a phrase "year-old" probably refers to age.
Entity present with a phrase in sentence "who turned" probably refers to age.
Entity1 is "also known as" Entity2 probably refers to alternate name.
Entity1, "nicknamed" Entity2 probably refers to alternate name.
Entity1 followed by phrase "is a citizen of" Entity2 probably refers to origin.
Entity followed by phrase "is a devout" Entity2 probably refers to religion.
Entity, followed by "a" Entity2"-based company" probably refers to city/state/country of headquarters.
If Entity1 and Entity2 are siblings then they are not parents of each other.
If Entity1 and Entity2 are spouses of each other then they are not parents of each other.

acyclicity making these DNs particularly easy to be learned. We refer the reader to previous work [14, 12] for more details and examples of the RDN model.

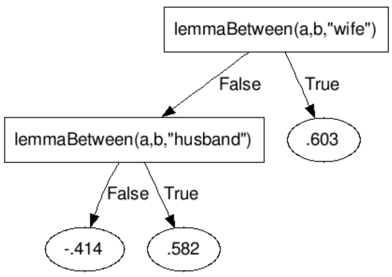

Fig. 2. Example regression tree for the *siblings* relation. This tree states that the weight for the relation being true is higher if either "husband" or "wife" appear between the entities.

In an RDN, typically, each distribution is represented by a relational probability tree (RPT) [15]. However, following previous work [12], we replace the RPT of each distribution with a set of relational regression trees [1] built in a sequential manner i.e., replace a single tree with a set of gradient boosted trees. This approach, RDN Boost, has been shown to have state-of-the-art results in learning RDNs. An simplified regression tree for the siblings relation is provided in Figure 2. Several boosted trees are learned for each relation and combined in ensemble fashion during inference.

3.4 Incorporating Human Advice

While most relational learning methods restrict the human to merely annotating the data, we go beyond and request the human for advice. The intuition is that we as humans read certain patterns and use them to deduce the nature of the relation between two entities present in the text. The goal of our work is to capture such mental patterns of the humans as advice to the learning algorithm. We modified the work of Odom et al. [18, 19] to learn RDNs in the presence of advice. The key idea is to explicitly represent advice in calculating gradients. This allows the system to trade-off between data and advice throughout the learning phase, rather than only consider advice in initial iterations. Advice, in particular, become influential in the presence of noisy or less amount of data.

A few sample advice rules in English (converted to FOL in RDN Boost) are presented in Table 3. Note that some of the rules are "soft" rules in that they are not true in many situations. Odom et al. [19] weigh the effect of the rules against the data and hence allow for partially correct rules.

4 Experiments and Results

Table 4. Relations The relations considered from TAC KBP. Columns indicate the number of training examples utilized – both human annotated (Gold) and weakly supervised (WS), when available – from TAC KBP 2014 and number of test examples from TAC KBP 2015. 10 relations describe person entities (*per*) while the last 4 describe organizations (*org*).

Relation	Gold	WS	Test
$per : age$	89	750	44
$per : alternateName$	28	x	18
$per : children$	89	x	23
$per : origin$	96	750	48
$per : otherFamily$	72	750	10
$per : parents$	71	750	30
$per : religion$	70	750	11
$per : siblings$	77	750	31
$per : spouse$	66	750	28
$per : title$	158	x	39
$org : cityHQ$	69	x	10
$org : countryHQ$	69	21	29
$org : dateFounded$	70	750	17
$org : foundedBy$	62	750	32

We now present our experimental evaluation. We considered 14 specific relations from two categories, *person* and *organization* from the TAC KBP competition. The relations considered are listed in the left column of Table 4. We utilize documents from KBP 2014 for training while utilizing a non-overlapping set of documents from the 2015 corpus for testing.

All RDN results presented are obtained from 5 different runs of the train and test sets to provide more robust estimates of accuracy[8]. We consider three standard metrics – area under the ROC curve, F-1 score and the recall at a certain precision. The train/test gold-standard sizes are provided in the table, including weakly supervised examples. Negative examples are created by randomly selecting paired entities in the same sentence (per relation and per run). We chose the precision as 0.66 since the fraction of positive examples to negatives is 1:2.

To analyze our system, we aimed to answer the following questions:

Q1: Do weakly supervised examples help construct better models?
Q2: Does joint learning help in some relations?
Q3: Are `word2vec` features more predictive than standard features?
Q4: Does advice improve performance compared to just learning from data?
Q5: Does our system perform competitively against a robust baseline?

[8] Please see [12] for standard settings. 25 trees were learned per relation per run, with maximum depth of 3 and advice learning rate of 0.25

4.1 Weak Supervision

Table 5. Weak Supervision Results comparing models trained with gold standard examples only (G) and models trained with gold standard and weakly supervised examples combined (G+WS).

Relation	AUC ROC		
	G	W	G+W
age	0.94	0.83	0.91
origin	0.88	0.69	0.77
otherFamily	0.88	0.88	0.93
parents	0.74	0.69	0.70
religion	0.77	0.70	0.80
siblings	0.82	0.72	0.74
spouse	0.86	0.86	0.76
countryHQ	0.79	0.60	0.79
dateFounded	0.87	0.81	0.84
foundedBy	0.85	0.61	0.70

To address **Q1**, we sought to analyze whether weakly supervised examples could substitute for a large gold-standard training set. Specifically, we evaluated 10 relations as show in Table 5. Based on experiments from [21], we utilized our knowledge-based weak supervision approach to provide positive examples, with a range of 4 to 8 rules for each relation. We compared three conditions: using (1) a large gold-standard training set (G), (2) a large weakly supervised data set (750 positive examples per relation) (W), and (3) using a small sample of 30 gold standard combined with 150 weakly supervised examples (G+W).

The results are presented in Table 5. With a few exceptions, a small set of seed gold standard examples combined with weakly supervised data mimic the results of a much larger (and time-consuming) gold-standard set of training data. The exceptions are cases where our knowledge-base approach struggles to find quality examples. In previous work, we showed that distant supervision may be better in cases like this where general world knowledge is difficult to encapsulate in rules [21]. Surprisingly, a large weak supervision set by itself does not seem to help learn better models for inferring relations in most cases. We hypothesize that the number of gold standard examples provided may be sufficient to learn RDN models. Thus **Q1** is answered equivocally, with weak supervision being able to supplement a small amount of gold examples.

4.2 Joint learning

To address our next question, we assessed our pipeline when learning relations independently (i.e., individually) versus learning relations jointly within the RDN, displayed in Table 6. Recall and F1 are omitted for conciseness – the conclusions are the same across all metrics. Joint learning appears to help in about half of the relations (8/14). Particularly, in person category, joint learning with gold standard outperforms their individual learning counterparts. This is due to the fact that some relations such as parents, spouse, siblings etc. are inter-related and learning them jointly indeed improves performance. Hence **Q2** can be answered affirmatively for connected relations.

Table 6. Joint Learning Results comparing relation models learned individually (IL) and jointly (JL).

Relation	AUC ROC IL	AUC ROC JL
age	0.93	0.93
alternateName	0.91	0.75
children	0.75	0.76
origin	0.86	0.89
otherFamily	0.88	0.89
parents	0.74	0.74
religion	0.72	0.79
siblings	0.79	0.80
spouse	0.86	0.87
title	0.90	0.89
cityHQ	0.74	0.73
countryHQ	0.75	0.79
dateFounded	0.87	0.86
foundedBy	0.83	0.86

Table 7. word2vec Results comparing models trained without (-w2v) and with `word2vec` features (+w2v).

Relation	AUC ROC -w2v	AUC ROC +w2v
age	0.94	0.94
alternateName	0.75	0.73
children	0.76	0.79
origin	0.88	0.86
otherFamily	0.88	0.88
parents	0.74	0.76
religion	0.77	0.79
siblings	0.82	0.79
spouse	0.86	0.82
title	0.89	0.90
cityHQ	0.73	0.73
countryHQ	0.79	0.78
dateFounded	0.87	0.88
foundedBy	0.85	0.84

4.3 word2vec

Table 7 shows the results of experiments comparing the RDN framework with and without `word2vec` features. We set the modes [22] such that the first argument to *isCosSimilar* is a '-' (i.e., existing) variable. We provide lists of candidate constants (on average a few dozen for each target relation) for the second argument using our knowledge of each target concept. For example, we include the words "father" and "mother" (for the *parent* relation) or "devout", "convert", and "follow" (*religion* relation). For the third argument, we utilize a threshold for cosine similarity of 0.70 (i.e., a word is considered to be similar to a keyword if their cosine similarity is above 0.70).

`word2vec` appears to largely have no impact on results. One possibility may be that this is due to a limitation in the depth of trees learned. Learning more and/or deeper trees may improve use of `word2vec` features. Moreover, our experiments were limited to a single threshold for similarity. Instead, we could provide a list of thresholds so that the learner could utilize different similarity thresholds for different contexts. **Q3** is answered cautiously in the negative, although future work could lead to improvements.

4.4 Advice

Table 8 shows the results of experiments that test the use of advice within the joint learning setting. The use of advice improves or matches the performance of using only joint learning. The key impact of advice can be mostly seen in the improvement of recall in several relations. It appears that in cases where there are not good quality examples, advice improves recall but in cases where there are already reasonable examples, advice does not improve the performance significantly. This is line with previous findings in other domains [24, 3, 5, 19]. As this claim warrants further investigation, Q4 can be answered optimistically.

4.5 RDN Boost vs RelationFactory

Table 8. Advice Results comparing models trained without (-Adv) and with advice (+Adv).

Relation	AUC ROC		Recall	
	-Adv	+Adv	-Adv	+Adv
age	0.93	0.93	0.56	0.74
alternateName	0.75	0.77	0.20	0.16
children	0.76	0.76	0.04	0.14
origin	0.89	0.88	0.86	0.82
otherFamily	0.89	0.90	0	0.06
parents	0.74	0.72	0.15	0.05
religion	0.79	0.81	0.51	0.56
siblings	0.80	0.81	0.04	0.00
spouse	0.87	0.85	0.06	0.04
title	0.89	0.90	0.16	0.07
cityHQ	0.73	0.74	0.26	0.28
countryHQ	0.79	0.77	0.61	0.62
dateFounded	0.86	0.86	0.20	0.05
foundedBy	0.86	0.84	0.24	0.25

RelationFactory (RF) [20] is an open-source system for performing relation extraction based on distantly supervised classifiers. It was the top system in the TAC KBP 2013 competition [23] and thus serves as a suitable baseline for our method. RF is very conservative in its responses, making it difficult to adjust the precision levels. To be most generous to RF, we present recall for all returned results. The AUC ROC, recall, and F1 scores of our system against RF are presented in Table 9. Inference in RF took approximately 15 minutes on a single CPU for the entire test set and 3 minutes for RDNs. Training and RDN took approximately 30 minutes per relation per run (RF is available pre-trained).

Based on the results, we can conclude for **Q5** that RDNs performs comparably, and often better than the state-of-the-art RelationFactory system. In particular, our method outperforms RelationFactory in AUC ROC across all relations. Recall is a mixed picture with both approaches showing some improvements – RDN outperforms in 6 relations while RelationFactory does so in 8. Note that in the instances where RDN provides superior recall, it does so with dramatic improvements (RF often returns 0 positives in these relations). F1 also shows RDN's superior performance, outperforming RF in most relations.

5 Conclusion

We presented our fully relational system utilizing Relational Dependency Networks for the Knowledge Base Population task. We demonstrated RDN's ability to effectively learn the relation extraction task, performing comparably (and often better) than the state-of-art RelationFactory system. Furthermore, we demonstrated the ability of RDNs to incorporate various concepts in a relational framework, including `word2vec`, human advice, joint learning, and weak

Table 9. RelationFactory (RF) vs RDN Values in bold indicate superiour performance against the alternative approach.

Relation	AUC ROC		Recall		F1	
	RF	RDN	RF	RDN	RF	RDN
age	0.64	**0.93**	0.28	**0.74**	0.44	**0.67**
alternateName	0.50	**0.77**	0.00	**0.16**	0	**0.10**
children	0.54	**0.76**	0.09	**0.14**	0.17	**0.28**
origin	0.50	**0.89**	0.00	**0.86**	0	**0.64**
otherFamily	0.56	**0.90**	**0.11**	0.06	**0.24**	0.22
parents	0.29	**0.74**	**0.33**	0.15	**0.50**	0.31
religion	0.50	**0.81**	0	**0.56**	0	**0.60**
siblings	0.13	**0.81**	**0.17**	0.00	0.29	0.29
spouse	0.57	**0.85**	**0.13**	0.04	0.23	**0.37**
title	0.67	**0.90**	**0.67**	0.07	**0.80**	0.54
cityHQ	0.38	**0.74**	**0.38**	0.28	**0.55**	0.41
countryHQ	0.57	**0.77**	0.14	**0.62**	0.25	**0.58**
dateFounded	0.67	**0.86**	**0.33**	0.05	**0.50**	0.46
foundedBy	0.20	**0.84**	**0.37**	0.25	0.54	**0.55**

supervision. While weak supervision did not significantly improve performance on its own, we demonstrate that our knowledge-base weak supervision approach is a viable alternative to the more popular distant supervision approach and can effectively substitute for a much larger (and expensive) gold-standard data set. Furthermore, while initial results show `word2vec` features do not improve accuracy on this task, our work does demonstrate a viable formulation for utilizing these features in a relation setting. Future directions include considering a larger number of relations, deeper features and finally, comparisons with more systems which have experienced recent success, such as DeepDive [17].

Acknowledgements Dileep Viswanathan, Jude Shavlik and Sriraam Natarajan gratefully acknowledge the support of the DARPA DEFT Program under the Air Force Research Laboratory (AFRL) prime contract no. FA8750-13-2-0039. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the DARPA, ARO, AFRL, or the US government.

References

1. Blockeel, H., Raedt, L.D.: Top-down induction of first-order logical decision trees. Artificial intelligence 101(1), 285–297 (1998)
2. Domingos, P., Lowd, D.: Markov Logic: An Interface Layer for AI. Morgan & Claypool, San Rafael, CA (2009)
3. Fung, G.M., Mangasarian, O.L., Shavlik, J.W.: Knowledge-based support vector machine classifiers. In: NIPS. pp. 01–09. MIT Press (2002)
4. Heckerman, D., Chickering, D., Meek, C., Rounthwaite, R., Kadie, C.: Dependency networks for inference, collaborative filtering, and data visualization. Journal of Machine Learning Research pp. 49–75 (2001)

5. Kunapuli, G., Odom, P., Shavlik, J., Natarajan, S.: Guiding an autonomous agent to better behaviors through human advice. In: ICDM. pp. 409–418 (2013)
6. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: ACL. pp. 55–60 (2014)
7. Meza-Ruiz, I., Riedel, S.: Jointly identifying predicates, arguments and senses using markov logic. In: NAACL-HLT. pp. 155–163 (2009)
8. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. Workshop at ICLR (2013)
9. Mikolov, T., Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. In: NAACL-HLT. pp. 746–751 (2013)
10. Min, B., Grishman, R., Wan, L., Wang, C., Gondek, D.: Distant supervision for relation extraction with an incomplete knowledge base. In: NAACL. pp. 777–782 (2013)
11. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: ACL. pp. 1003–1011 (2009)
12. Natarajan, S., Khot, T., Kersting, K., Gutmann, B., Shavlik, J.: Boosting relational dependency networks. In: ILP (2010)
13. Natarajan, S., Picado, J., Khot, T., Kersting, K., Re, C., Shavlik, J.: Effectively creating weakly labeled training examples via approximate domain knowledge. In: ILP (2014)
14. Neville, J., Jensen, D.: Relational dependency networks. In: Introduction to Statistical Relational Learning. The MIT Press (2007)
15. Neville, J., Jensen, D., Friedland, L., Hay, M.: Learning relational probability trees. In: SIGKDD. pp. 625–630 (2003)
16. Niu, F., Ré, C., Doan, A., Shavlik, J.W.: Tuffy: Scaling up statistical inference in Markov logic networks using an RDBMS. VLDB 4(6), 373–384 (2011)
17. Niu, F., Zhang, C., Ré, C., Shavlik, J.W.: Deepdive: Web-scale knowledge-base construction using statistical learning and inference. VLDS 12, 25–28 (2012)
18. Odom, P., Bangera, V., Khot, T., Page, D., Natarajan, S.: Extracting adverse drug events from text using human advice. In: Artificial Intelligence in Medicine. pp. 195–204 (2015)
19. Odom, P., Khot, T., Porter, R., Natarajan, S.: Knowledge-based probabilistic logic learning. In: AAAI. pp. 3564–3570 (2015)
20. Roth, B., Barth, T., Chrupala, G., Gropp, M., Klakow, D.: Relationfactory: A fast, modular and effective system for knowledge base population. In: EACL. pp. 89–92 (2014)
21. Soni, A., Viswanathan, D., Pachaiyappan, N., Natarajan, S.: A comparison of weak supervision methods for knowledge base construction. In: Automated Knowledge Base Construction (AKBC) Workshop at NAACL (2016)
22. Srinivasan, A.: The aleph manual (2001), `http://www.comlab.ox.ac.uk/oucl/~research/areas/machlearn/Aleph/`
23. Surdeanu, M.: Overview of the TAC 2013 knowledge base population evaluation: English slot filling and temporal slot filling. In: Text Analysis Conference (2013)
24. Towell, G., Shavlik, J.: Knowledge-based artificial neural networks. Artificial Intelligence 70(1-2), 119–165 (1994)
25. Viswanathan, V., Rajani, N.F., Bentor, Y., Mooney, R.J.: Stacked ensembles of information extractors for knowledge-base population. In: ACL. pp. 177–187 (2015)