Deep Neural Networks with Massive Learned Knowledge

Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, Eric P. Xing
School of Computer Science
Carnegie Mellon University
{zhitingh,zichaoy,rsalakhu,epxing}@cs.cmu.edu

Abstract

Regulating deep neural networks (DNNs) with human structured knowledge has shown to be of great benefit for improved accuracy and interpretability. We develop a general framework that enables learning knowledge and its confidence jointly with the DNNs, so that the vast amount of fuzzy knowledge can be incorporated and automatically optimized with little manual efforts. We apply the framework to sentence sentiment analysis, augmenting a DNN with massive linguistic constraints on discourse and polarity structures. Our model substantially enhances the performance using less training data, and shows improved interpretability. The principled framework can also be applied to posterior regularization for regulating other statistical models.

1 Introduction

Deep neural networks (DNNs) have achieved remarkable success in a large variety of application domains (Krizhevsky et al., 2012; Hinton et al., 2012; Bahdanau et al., 2014). However, the powerful end-to-end learning comes with limitations, including the requirement on massive amount of labeled data, uninterpretability of prediction results, and difficulty of incorporating human intentions and domain knowledge.

To alleviate these drawbacks, recent work has focused on training DNNs with extra domain-specific features (Collobert et al., 2011), combining oracle similarity constraints (Karaletsos et al., 2016), modeling output correlations (Deng et al., 2014), and others. Recently, Hu et al. (2016) proposed a general distillation framework that transfers knowledge expressed as first-order logic (FOL) rules into neural networks, where FOL constraints are integrated via posterior regularization (Ganchev et al., 2010). Despite the intuitiveness of FOL rules and the impressive performance in various tasks, the approach, as with the previous posterior constraint methods (Ganchev et al., 2010; Liang et al., 2009; Zhu et al., 2014), has been limited to simple *a priori* fixed constraints with manually selected weights, lacking the ability of inducing and adapting abstract knowledge from data. This issue is further exacerbated in the context of regulating DNNs that map raw data directly into the label space, leaving a huge semantic gap in between, and making it unfeasible to express rich human knowledge built on the intermediate abstract concepts.

In this paper, we introduce a generalized framework which enables a learning procedure for knowledge representations and their weights jointly with the regulated DNN models. This greatly extends the applicability to massive structures in diverse forms, such as structured models and soft logic rules, facilitating practitioners to incorporate rich domain expertise and fuzzy constraints. Specifically, we propose a *mutual* distillation method that iteratively transfers information between DNN and structured knowledge, resulting in effective integration of the representation learning capacity of DNN and the generalization power of structured knowledge. Our method does not require additional supervision beyond raw data-labels for knowledge learning.

We present an instantiation of our method in the task of sentence sentiment analysis. We aug-

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1670–1679,
Austin, Texas, November 1-5, 2016.

ment a base convolutional network with linguistic knowledge that encourages coherent sentiment transitions across the clauses in terms of discourse relations. All uncertain modules, such as clause relation and polarity identification, are automatically learned from data, freeing practitioners from exhaustive specification. We further improve the model by integrating thousands of soft word polarity and negation rules, with their confidence directly induced from the data.

Trained with only sentence level supervisions, our model substantially outperforms plain neural networks learned from both sentence and clause labels. Our method also shows enhanced generalization on limited data size, and improved interpretability of predictions.

Our work enjoys general versatility on diverse types of structured knowledge and neural architectures. The principled knowledge and weight learning approach can also be applied to the posterior constraint frameworks (Ganchev et al., 2010; Liang et al., 2009) for regulating other statistical models.

2 Related Work

Deep Networks with Structured Knowledge Combining the powerful deep neural models with structured knowledge has been of increasing interest to enhance generalization and improve interpretability (Li et al., 2015; Deng et al., 2014; Johnson et al., 2016). Recently, Hu et al. (2016) proposed to transfer logical knowledge information into neural networks with diverse architectures (e.g., convolutional networks and recurrent networks). They developed an iterative distillation framework that trains the neural network to emulate the predictions of a "teacher" model which is iteratively constructed by imposing posterior constraints on the network. The framework has shown to be effective in regulating different neural models. However, the method has required fixed constraints and manually specified weights, making it unsuitable to incorporate large amount of fuzzy human intuitions where adaptation to data is necessary to obtain meaningful knowledge representations.

The limitation is in fact shared with the general-purpose posterior regularization methods (Ganchev et al., 2010; Liang et al., 2009; Zhu et al., 2014). Though attempts have been made to learn the constraint weights from additional supervisions (Mei et al., 2014) or for tractability purposes (Steinhardt and Liang, 2015), learning and optimizing knowledge expressions jointly with the regulated models from data is still unsolved, and critically restricting the application scope.

Sentiment Analysis Sentence level sentiment classification is to identify the sentiment polarity (e.g., positive or negative) of a sentence (Pang and Lee, 2008). Recently, a number of neural models have been developed and achieved new levels of performance (Kim, 2014; Socher et al., 2013; Lei et al., 2015). Despite the impressive success, most of the existing neural network approaches require large amount of labeled data while encoding very limited linguistic knowledge, making them inefficient to handle sophisticated linguistic phenomena, such as contrastive transitions and negations (Choi and Cardie, 2008; Bhatia et al., 2015).

Hu et al. (2016) combines a neural network with a logic rule that captures contrastive sense by observing the word "but" in a sentence. However, such simple deterministic rules suffer from limited generality and robustness. This paper develops a new sentiment neural model that combines a large diverse set of linguistic knowledge through our enhanced framework. Our method efficiently captures complex linguistic patterns from limited data, and yields highly interpretable predictions.

3 Mutual Distillation

This section introduces the proposed framework that enables joint learning of knowledge components and their weights with the neural network models. In particular, we generalize the one-sided distillation method of (Hu et al., 2016) (section 3.1), and propose to mutually transfer information between the neural network and the structured constraints for effective knowledge learning (section 3.2), and optimize the weights by considering jointly all components (section 3.3).

We consider input variable $\boldsymbol{x} \in \mathcal{X}$ and target variable $\boldsymbol{y} \in \mathcal{Y}$. For clarity we focus on classification where $\boldsymbol{y}$ is a one-hot encoding of the class labels, though our method also applies to other contexts. Let $(\boldsymbol{X}, \boldsymbol{Y})$ denote a set of instances of $(\boldsymbol{x}, \boldsymbol{y})$.

A neural network defines a conditional probability $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ parameterized by $\boldsymbol{\theta}$. We will omit the subscript θ when there is no ambiguity.

3.1 Network Learning with Knowledge Distillation

We first review the iterative distillation method (Hu et al., 2016) that transfers structured knowledge into neural networks. Consider constraint functions $f_l \in \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, indexed by l, that encode the knowledge and we want to satisfy (i.e., maximize by optimizing the predictions $\boldsymbol{y}$) with confidence weights $\lambda_l \in \mathbb{R}$. Given the current state of the neural network parameters $\boldsymbol{\theta}$ at each iteration, a structure-enriched teacher network q is obtained by solving

$$\min_{q\in\mathcal{P}} \mathrm{KL}(q(\boldsymbol{Y})\|p_\theta(\boldsymbol{Y}|\boldsymbol{X})) - C\sum\nolimits_l \lambda_l \mathbb{E}_q[f_l(\boldsymbol{X},\boldsymbol{Y})], \tag{1}$$

where $\mathcal{P}$ denotes the appropriate distribution space; and C is the regularization parameter. Problem (1) is convex and has a closed-form solution

$$q^*(\boldsymbol{Y}) \propto p_\theta(\boldsymbol{Y}|\boldsymbol{X}) \exp\left\{C\sum\nolimits_l \lambda_l f_l(\boldsymbol{X},\boldsymbol{Y})\right\}, \tag{2}$$

whose normalization term can be calculated efficiently according to how the constraints factorize (Hu et al., 2016). The neural network p_θ at iteration t is then updated with a distillation objective (Hinton et al., 2015) that balances between imitating soft predictions of teacher q and predicting true hard labels:

$$\begin{aligned}\boldsymbol{\theta}^{(t+1)} = \arg\min_{\theta\in\Theta} \frac{1}{N}\sum_{n=1}^{N} &(1-\pi)\ell(\boldsymbol{y}_n, \boldsymbol{\sigma}_\theta(\boldsymbol{x}_n)) \\ &+ \pi\ell(\boldsymbol{s}_n^{(t)}, \boldsymbol{\sigma}_\theta(\boldsymbol{x}_n)),\end{aligned} \tag{3}$$

where ℓ denotes the loss function (e.g., cross entropy loss for classification); $\boldsymbol{\sigma}_\theta(\boldsymbol{x})$ is the softmax output of p_θ on $\boldsymbol{x}$; $\boldsymbol{s}_n^{(t)}$ is the soft prediction vector of q on training point $\boldsymbol{x}_n$ at iteration t; N is the training size; and π is the imitation parameter calibrating the relative importance of the two objectives. The training procedure iterates between Eq.(2) and Eq.(3), resulting in the richly structured teacher model q and the knowledge distilled student network p. While q generally provides better accuracy, p is more lightweight and applicable to many different contexts (Hu et al., 2016; Liang et al., 2008).

In (Hu et al., 2016), the constraint $f_l(\boldsymbol{X},\boldsymbol{Y})$ has been limited to be of the form $r_l(\boldsymbol{X},\boldsymbol{Y}) - 1$, where r_l is an FOL function yielding truth values in $[0, 1]$, and is required to be fully-specified *a priori* and fixed throughout the training. Besides, the constraint weight λ_l has to be manually selected. This severely deviates from the characters of human knowledge which is usually abstract, fuzzy, built on high-level concepts (e.g., discourse relations, visual attributes) as opposed to low-level observations (e.g., word sequences, image pixels), and thus incomplete in the sense of end-to-end learning that maps raw input directly into target space of interest. This necessitates expressing structured knowledge allowing some modules unknown and induced automatically from observations.

3.2 Knowledge Learning

To substantially extend the scope of knowledge used in the framework, we introduce learnable modules $\boldsymbol{\phi}$ in the knowledge expression denoted as f_ϕ. The module $\boldsymbol{\phi}$ is general, and can be, e.g., free parameters of structured metrics, or dependency structures over semantic units. We assume f_ϕ can be optimized in terms of $\boldsymbol{\phi}$ against a given objective (e.g., through gradient descent for parameter updating). We aim to learn the knowledge by determining $\boldsymbol{\phi}$ from data.

For clarity we consider one knowledge constraint and omit the index l. We further assume the constraint factorizes over data instances. Note that our method can straightforwardly be applied to the case of multiple constraints and constraints spanning multiple instances. As any meaningful knowledge is expected to be consistent with the observations, a straightforward way is then to directly optimize against the training data: $\boldsymbol{\phi}^* = \arg\max_{\boldsymbol{\phi}} \frac{1}{N}\sum_n f_\phi(\boldsymbol{x}_n, \boldsymbol{y}_n)$, and insert the resulting f_{ϕ^*} in Eq.(1) for subsequent steps. However, such a pipelined method fails to establish interactions between the knowledge and network learning, and can lead to a sub-optimal system, as shown in our experiments.

To address this, we inspect the posterior regularization objective in Eq.(1), and write it in an analogous form to the variational free energy of some model evidence. Specifically, let $\log h_\phi(\boldsymbol{X},\boldsymbol{Y}) \triangleq C\lambda f_\phi(\boldsymbol{X},\boldsymbol{Y})$, then the objective can be written as

$$-\sum_{\boldsymbol{Y}} q(\boldsymbol{Y}) \log \frac{p(\boldsymbol{Y}|\boldsymbol{X}) h_\phi(\boldsymbol{X},\boldsymbol{Y})}{q(\boldsymbol{Y})}. \tag{4}$$

Intuitively, we can view the output distribution of the neural network $p(\boldsymbol{Y}|\boldsymbol{X})$ as a prior distribution over the labels, while considering $h_\phi(\boldsymbol{X}, \boldsymbol{Y})$ as defining a "likelihood" metric w.r.t the observations, making the objective analogous to a (negative) variational lower bound of the respective "model". This naturally inspires an EM-type algorithm (Neal and Hinton, 1998) to optimize relevant parameters and improve the "evidence": the E-step optimizes over q, yielding Eq.(2); and the M-step optimizes over $\boldsymbol{\phi}$. Further incorporating the true training labels with balancing parameter π', we obtain the update for $\boldsymbol{\phi}$:

$$\begin{aligned}\boldsymbol{\phi}^{(t+1)} = \arg\max_{\phi \in \Phi} \frac{1}{N} \sum_{n=1}^{N} &(1-\pi') h_\phi(\boldsymbol{x}_n, \boldsymbol{y}_n) \\ &+ \pi' \mathbb{E}_{q^{(t)}(\boldsymbol{y})}[h_\phi(\boldsymbol{x}_n, \boldsymbol{y})]\end{aligned} \tag{5}$$

The update rule resembles the distillation objective for learning parameters $\boldsymbol{\theta}$ in Eq.(3). Indeed, the expectation term in Eq.(5) in effect optimizes h_ϕ on examples labeled by $q(\boldsymbol{y})$, i.e., forcing the knowledge function to mimic the predictions of the teacher model and distill encoded information. Thus, besides transferring from structured knowledge to a neural model by Eq.(3), we now further bridge from the neural network to the knowledge constraints for joint learning and better integrating the best of both worlds. We call our framework with the symmetric objectives as *mutual distillation*. In fact, we can view Eq.(4) as a single joint objective and we are alternating optimization of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, resulting in the update rules in Eq.(3) and Eq.(5) with the supervised loss terms included, respectively (and with the loss function in Eq.(3) being cross-entropy loss).

Additionally, the resemblance of the two objectives indicates that we can readily translate the successful neural learning method to knowledge learning. For instance, the expectation term in Eq.(5), as the second loss term in Eq.(3), can be evaluated on rich unlabeled data in addition to labeled examples, enabling semi-supervised learning which has shown to be useful (Hu et al., 2016). Empirical studies show superiority of the proposed method over several potential alternatives (section 5).

3.3 Weight Learning

Besides optimizing the knowledge representations, we also aim to automate the selection of constraint weights by learning from data. This would enable us to incorporate massive amount of noisy knowledge, without the need to worry about the confidence which is usually unfeasible to set manually.

As the constraint weights serve to balance between the different components of the whole framework, we learn the weights by optimizing the regularized joint model q (see Eq.(2)):

$$\boldsymbol{\lambda}^{(t+1)} = \arg\max_{\boldsymbol{\lambda} \geq 0} \frac{1}{N} \sum_{n=1}^{N} q_\lambda(\boldsymbol{y}_n) \tag{6}$$

This is also validated in the view of regularized Bayes (Zhu et al., 2014) where q is a generalized posterior function by regularizing the standard posterior p (see Eq.(1)). Although here, we omit the Bayesian treatment of the weights $\boldsymbol{\lambda}$ and instead optimize them directly to find the posterior. It is straightforward to impose priors over $\boldsymbol{\lambda}$ to encode preferences. In practice, Eq. (6) can be carried out through gradient descent.

The training procedure of the proposed mutual distillation is summarized in Algorithm 1.

Algorithm 1 Mutual Distillation

```
Input: Training data $\mathcal{D} = \{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^N$,
       Initial knowledge constraints $\mathcal{F} = \{f_{\phi,l}\}_{l=1}^L$,
       Initial neural network $p_\theta$,
       Parameters: $\pi$, $\pi$' – imitation parameters
                   $C$ – regularization parameters
Initialize neural network parameters $\boldsymbol{\theta}$
Initialize knowledge parameters $\boldsymbol{\phi}$ and weights $\boldsymbol{\lambda}$
while not converged do
    Sample a minibatch $(\boldsymbol{X}, \boldsymbol{Y}) \subset \mathcal{D}$
    Build the teacher model $q$ with Eq.(2) and Eq.(6)
    Update $p_\theta$ with distillation objective Eq.(3)
    Update $f_l$ $(l = 1, \dots, L)$ with distillation objective Eq.(5)
end while
Output: Learned network $p$, knowledge modules $\mathcal{F}$, and
    the joint teacher network $q$
```

4 Sentiment Classification

This section provides a concrete instance of our general framework in the task of sentence sentiment analysis. We augment a base convolutional network with a large diverse set of linguistic knowledge, including 1) sentiment transition structure for coherent multi-level prediction, 2) conjunction word rules

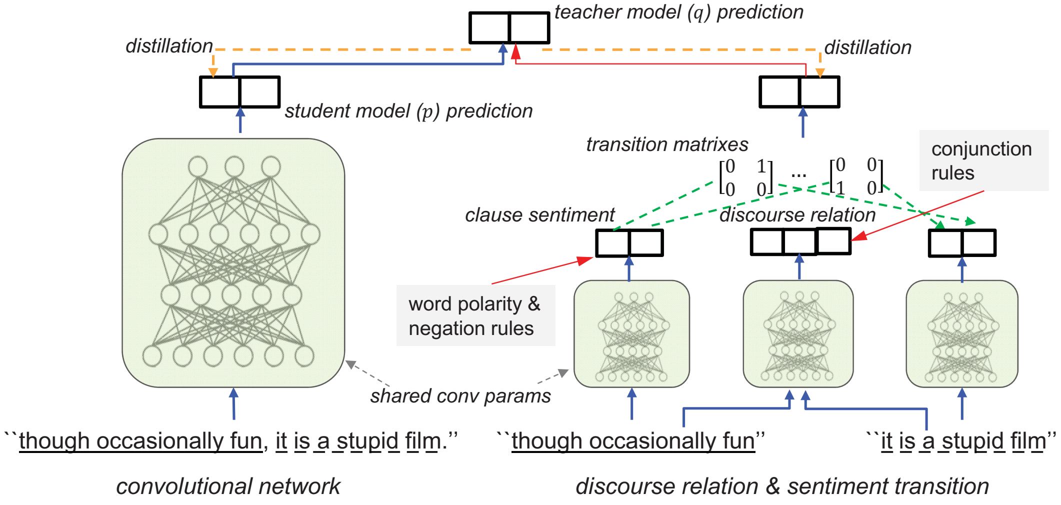
teacher model (q) prediction
distillation
distillation
student model (p) prediction
transition matrixes
[0 1; 0 0] ... [0 0; 1 0]
conjunction rules
clause sentiment
discourse relation
word polarity & negation rules
shared conv params
``though occasionally fun, it is a stupid film.''
``though occasionally fun''
``it is a stupid film''
convolutional network
discourse relation & sentiment transition

Figure 1: Our sentiment classification model. The left part is the base convolutional network over sentences, and the right part is the knowledge component over clauses. Blue arrows denote neural feed-forwards; red arrows denote knowledge incorporation steps; and the orange dashed arrows denote the distillation processes. The convolutional parameters are shared across all the networks.

for improving discourse relation identification, and 3) word polarity rules for tackling negations. These knowledge structures are fulfilled with neural network modules that are learned jointly within our framework. The resulting model efficiently captures sophisticated linguistic patterns from limited data, and produces interpretable predictions.

Figure 1 shows an overview of our model. We assume binary sentiment labels (i.e., positive-1 and negative-0). The left part of the figure is the base neural network for sentence classification. Since our framework is agnostic to the neural architecture, we can use any off-the-shelf neural models such as convolutional network and recurrent network. Here we choose the simple yet effective convolutional network proposed in (Kim, 2014). The network takes as input the word embedding vectors of a given sentence, and extracts feature maps with a convolutional layer followed by max-over-time pooling. A final fully-connected layer with softmax activation transforms the extracted features into a prediction vector.

We next introduce the three types of domain knowledge, which leverage rich fine-grained level structures, from clauses to words, to guide sentence level prediction. The clause segmentation of sentences is obtained using the public Stanford parser [1].

Sentiment transition by discourse relation Discourse structures characterize how the clauses (i.e., discourse units) of a sentence are connected with each other and thereby provide clues for coherent sentence and clause labeling. Instead of using standard general-purpose discourse relation system, we define three types of relations between adjacent clauses (denoted as c_i and c_{i+1}) specific to sentiment change, namely, *consistent* (c_i and c_{i+1} have the same polarity), *contrastive* (c_{i+1} opposes c_i and is the main part), and *concessive* (c_{i+1} opposes c_i and is secondary). The relations also indicate the connections between clauses and the whole sentence. For instance, a contrastive relation typically indicates c_{i+1} has the same polarity with the full sentence (we reasonably assume a sentence has contrastive sense in at most one position). To encode these dependencies we define sentiment *transition matrices* conditioned on discourse relation r and sentence polarity y, denoted as $M_{r,y}$. For instance, given $r =$ *contrastive* and $y = 0$, we expect the sentiment change between two adjacent clauses to follow

$$M_{r=\textit{contrastive},y=0} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \tag{7}$$

i.e., transiting from positive polarity of c_i to negative of c_{i+1}. We list all transition matrices in supplement.

We now design a constraint on sentence predictions leveraging the above knowledge. Using the identification modules presented shortly, we first get the discourse relation probabilities $\boldsymbol{p}^r_{i,i+1}$ as well as

[1] http://nlp.stanford.edu/software/openie.html

the sentiment polarity probabilities $\boldsymbol{p}^c_i$ and $\boldsymbol{p}^c_{i+1}$ of adjacent clauses (c_i, c_{i+1}). For a given sentence label y_s, we then compute the expected transition matrix at each position by $\bar{M}_{i,y_s} = \mathbb{E}_{\boldsymbol{p}^r_{i,i+1}}[M_{r,y_s}]$. The value of the constraint function on $y = y_s$ is then defined as the probability of the most likely clause polarity configuration according to the clause predictions $\boldsymbol{p}^c_\cdot$ and the averaged transitions $\bar{M}_{\cdot,y_s}$:

$$f^{st}(\boldsymbol{x}, y_s) = \max_{\boldsymbol{a}\in\{0,1\}^m} \prod\nolimits_i \boldsymbol{p}^r_{i,a_i} \cdot \bar{M}_{i,y_s,a_i a_{i+1}}, \qquad (8)$$

where $\boldsymbol{a}$ is the polarity configuration and m is the number of clauses. We use the Viterbi algorithm for efficient computation.

We need the clause relation and polarity probabilities $\boldsymbol{p}^r$ and $\boldsymbol{p}^c$, which are unfeasible to identify from raw text with only simple deterministic rules. We apply a convolutional network for each module, with similar network architectures to the base network (we describe details in the supplement). For efficiency, we tie the convolutional parameters across all the networks, while leaving the parameters of the fully-connected layers to be learned individually.

Conjunction word rules We enhance the discourse relation neural network with robust clues from explicit discourse connectives (e.g., "but", "and", etc.) that occur in the sentence. In particular, we collect a set of conjunction words (listed in the supplement) and specify a rule constraint for each of them. For instance, the conjunction "and" results in the following constraint function:

$$f^{rel}(c_i, c_{i+1}, r) = (\mathbf{1}_{\text{and}}(c_i, c_{i+1}) \Rightarrow r = \textit{consistent}),$$

where $\mathbf{1}_{\text{and}}(c_i, c_{i+1})$ is an indicator function that takes 1 if the two clauses are connected by "and", and 0 otherwise. Note that these rules are soft, with the confidence weights learned from data. We use the regularized joint model over the base discourse network for predicting the relations.

Negation and word polarity rules Negations reverse the polarity of relevant statements. Identifying negation sense has been a challenging problem for accurate sentiment prediction. We address this by incorporating rich lexicon rules at the clause level. That is, if a polarity-carrying word (e.g., "good") occurs in the scope of a negator (e.g., "not"), then the sentiment prediction of the clause is encouraged to be the opposite polarity. We specify one separate rule for each polarity-carrying word from public lexicons (see the supplement), e.g.,

$$f^{lex}(c_i, y_c) = \big(\mathbf{1}_{\text{good}}(c_i) \Rightarrow y_c = \textit{negative}\big), \qquad (9)$$

where $\mathbf{1}_{\text{good}}(c_i)$ is an indicator function that takes 1 if word "good" occurs in a negation scope in the clause text, and 0 otherwise. This results in over 3,000 rules, and our automated weight optimization frees us from manually selecting the weights exhaustively. We define the negation scope to be the 4 words following a negator (Choi and Cardie, 2008).

Though polarities of single words can be brittle features for determining the sentiment of a long statement due to complex semantic compositions, they are more robust and effective at the level of clauses which are generally short and simple. Moreover, inaccurate rules will be downplayed through the weight learning procedure.

We have presented our neural sentiment model. We tackle several long-standing challenges by directly incorporating linguistic knowledge. Comparing to previous work that designs various neural architectures and relies on substantial annotations for specific issues (Socher et al., 2013; Bhatia et al., 2015), our knowledge framework is more straightforward, interpretable, and general, while still preserving the power of neural methods.

Notably, even with several additional components to be learned for knowledge representation, our method does not require extra supervision signals beyond the raw sentence-labels, making our framework generally applicable to many different tasks (Neelakantan et al., 2016).

The sentiment transition knowledge is expressed in the form of structured model with features extracted using neural networks. Though apparently similar to recent deep structured models such as neural-CRFs (Durrett and Klein, 2015; Ammar et al., 2014; Do et al., 2010), ours is different since we parsimoniously extract features that are necessary for precise and efficient knowledge expression, as opposed to neural-CRFs that learn as rich representations as possible for final prediction.

5 Experiments

We evaluate our method on the widely-used sentiment classification benchmarks. Our knowledge

		Model	Accuracy (%)
sentences	1	CNN (Kim, 2014)	86.6
	2	CNN+REL	q: 87.8; p: 87.1
	3	CNN+REL+LEX	q: **88.0**; p: **87.2**
sentences	4	MC-CNN (Kim, 2014)	86.8
	5	Tensor-CNN (Lei et al., 2015)	87.0
	6	CNN+But-q (Hu et al., 2016)	87.1
+phrases	7	CNN (Kim, 2014)	87.2
	8	Tree-LSTM (Tai et al., 2015)	88.0
	9	MC-CNN (Kim, 2014)	88.1
	10	CNN+But-q (Hu et al., 2016)	89.2
	11	MVCNN (Yin and Schutze, 2015)	89.4

Table 1: Classification performance on SST2. The top and second blocks use only sentence-level annotations for training, while the bottom block uses both sentence- and phrases-level annotations. We report the accuracy of both the regularized teacher model q and the student model p after distillation.

enriched model significantly outperforms plain neural networks. We obtain even higher improvements with limited data sizes. Comparison with extensive other potential knowledge learning methods shows the effectiveness of our framework. Our model also shows improved interpretability.

5.1 Setup

Datasets Two classification benchmarks are used: 1) Stanford Sentiment Treebank-2 (**SST2**) (Socher et al., 2013) is a binary classification dataset that consists of 6920/872/1821 moview review sentences in the train/dev/test sets, respectively. Besides sentence-level annotations, the dataset also provides exhaustive gold-standard labels at fine-grained levels, from clauses to phrases. The resulting full training set includes 76,961 labeled instances. We train our model using only the *sentence-level* annotations, and compare to baselines learned from either training set. 2) Customer Reviews (**CR**) (Hu and Liu, 2004) consists of 3,775 product reviews with positive and negative polarities. Following previous work we use 10-fold cross-validation.

Model configurations We evaluate two variants of our model: CNN+REL leverages the knowledge of sentiment transition and discourse conjunctions, and CNN+REL+LEX additionally incorporates the negation lexicon rules.

Throughout the experiments we set the regularization parameter to $C = 10$. The imitation parameters π and π' decay as $\pi^{(t)} = \pi'^{(t)} = 0.9^t$ where t is the iteration number (Bengio et al., 2015; Hu et al., 2016). For the base neural network, we choose the "non-static" version from (Kim, 2014) and use the same configurations.

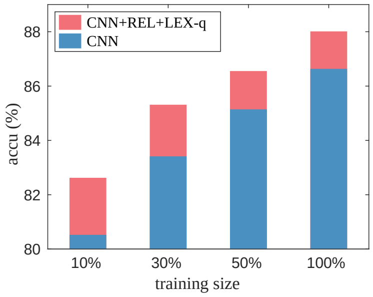

Figure 2: Performance with varying sizes of training examples.

5.2 Classification Results

Table 1 shows the classification performance on the SST2 dataset. From rows 1-3 we see that our proposed sentiment model that integrates the diverse set of knowledge (section 4) significantly outperforms the base CNN (Kim, 2014). The improvement of the student network p validates the effectiveness of the iterative mutual distillation process. Consistent with the observations in (Hu et al., 2016), the regularized teacher model q provides further performance boost, though it imposes additional computational overhead for explicit knowledge representations. Note that our models are trained with only sentence-level annotations. Compared with the baselines trained in the same setting (rows 4-6), our model with the full knowledge, CNN+REL+LEX, performs the best. CNN+But-q (row 6) is the base CNN augmented with a logic rule that identifies contrastive sense through explicit occurrence of word "but" (section 3.1) (Hu et al., 2016). Our enhanced framework enables richer knowledge and achieves much better performance.

Our method further outperforms the base CNN that is additionally trained with dense phrase-level annotations (row 7), showing improved generalization of the knowledge-enhanced model from limited data. Figure 2 further studies the performance with varying training sizes. We can clearly observe that the incorporated knowledge tends to offer higher improvement with less training data. This property can be particularly desirable in applications of structured predictions where manual annotations are expensive while rich human knowledge is available.

	Model	Accuracy (%)
1	CNN (Kim, 2014)	84.1±0.2
2	CNN+REL	q: 85.0±0.2; p: 84.7±0.2
3	CNN+REL+LEX	q: **85.3±0.3**; p: **85.0±0.2**
4	MC-CNN (Kim, 2014)	85.0
5	Bi-RNN (Lai et al., 2015)	82.6
6	CRF-PR (Yang and Cardie, 2014)	82.7
7	AdaSent (Zhao et al., 2015)	**86.3**

Table 2: Classification performance on the CR dataset. We report the average accuracy±one standard deviation with 10-fold CV. The top block compares the base CNN (row 1) with the knowledge-enhanced CNNs by our framework.

Table 2 shows model performance on the CR dataset. Our model again surpasses the base network and several other competitive neural methods by a large margin. Though falling behind AdaSent (row 7) which has a more specialized and complex architecture than standard convolutional networks, the proposed framework indeed is general enough to apply on top of it for further enhancement.

To further evaluate the proposed mutual distillation framework for learning knowledge, we compare to an extensive set of other possible knowledge optimization approaches. Table 3 shows the results. In row 2, the "opt-joint" method optimizes the regularized joint model of Eq.(2) directly in terms of both the neural network and knowledge parameters. Row 3, "opt-knwl-pipeline", is an approach that first optimizes the standalone knowledge component and then inserts it into the previous framework of (Hu et al., 2016) as a fixed constraint. Without interaction between the knowledge and neural network learning, the pipelined method yields inferior results. Finally, rows 4-5 display a method that adapts the knowledge component at each iteration by optimizing the joint model q in terms of the knowledge parameters. We report the accuracy of both the student network p (row 4) and the joint teacher network q (row 5), and compare with our method in row 6 and 7, respectively. We can see that both models performs poorly, achieving the accuracy of only 68.6% for the knowledge component, similar to the accuracy achieved by the "opt-joint" method.

	Model	Accuracy (%)
1	CNN (Kim, 2014)	86.6
2	opt-joint	86.9 (68.8)
3	opt-knwl-pipeline	86.7 (70.4)
4	opt-joint-iterative-p	86.9
5	opt-joint-iterative-q	87.6 (68.6)
6	mutual-p	87.2
7	mutual-q	**88.0** (72.5)

Table 3: Comparisons between our mutual distillation (rows 4-5) and other knowledge optimization methods, on SST2. See the text for details. The numbers in parentheses are the accuracy of the learned knowledge component (Figure 1, right part) if we take it as a standalone classifier. All knowledge is used.

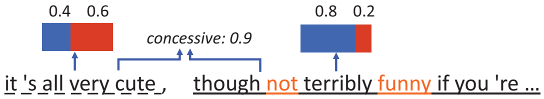
0.4
0.6
concessive: 0.9
0.8
0.2
it 's all very cute , though not terribly funny if you 're ...

Figure 3: An example sentence and the results of the learned knowledge modules applied on it. Red denotes *positive*, and blue denotes *negative*. The snippet "not ... funny" triggers the negation rule.

enough, good, strong, engaging, great	awful, loses, fake doubt, bad

Table 4: The top 5 positive (left) and negative (right) words with the largest weights of the negation rules.

In contrast, our mutual distillation framework offers the best performance. Table 3 shows that the knowledge component as a standalone classifier does not achieve high accuracy (the numbers in parentheses). As discussed in section 4, this is because of the parsimonious formulation for the precise knowledge expression, while leaving the expressive base NN to extract rich representations. The enhanced performance of the combination indicates complementary effects of the two parts.

5.3 Qualitative Analysis

Our model not only provides better classification performance, but also shows improved interpretability due to the learned structured knowledge representation. Figure 3 illustrates an example sentence from test set. We see that the clause sentiments as well as the discourse relation are correctly captured. The negation rule of "not ... funny" (Eq.(9)) also helps to identify the right polarity.

Table 4 lists the top-5 positive and negative words that are most confident for the negation rules, providing insights into the linguistic norms in the movie review context.

6 Conclusion

In this paper we have developed a framework that learns structured knowledge and its weights for regulating deep neural networks through mutual distillation. We instantiated our framework for the sentiment classification task. Using massive learned linguistic knowledge, our neural model provides substantial improvements over many of the existing approaches, especially in the limited data setting. In the future work, we plan to apply our framework to other text and vision applications.

Acknowledgments

We thank the anonymous reviewers for their valuable comments. This work is supported by NSF IIS1218282, NSF IIS1447676, Air Force FA8721-05-C-0003.

References

Waleed Ammar, Chris Dyer, and Noah A Smith. 2014. Conditional random field autoencoders for unsupervised structured prediction. In *Proc. of NIPS*, pages 3311–3319.

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

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proc. of NIPS*, pages 1171–1179.

Parminder Bhatia, Yangfeng Ji, and Jacob Eisenstein. 2015. Better document-level sentiment analysis from rst discourse parsing. In *Proc. of EMNLP*.

Yejin Choi and Claire Cardie. 2008. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proc. of EMNLP*, pages 793–801. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537.

Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. 2014. Large-scale object classification using label relation graphs. In *ECCV 2014*, pages 48–64. Springer.

Trinh Do, Thierry Arti, et al. 2010. Neural conditional random fields. In *Proc. of AISTATS*, pages 177–184.

Greg Durrett and Dan Klein. 2015. Neural CRF parsing.

Kuzman Ganchev, Joao Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *JMLR*, 11:2001–2049.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proc. of KDD*, pages 168–177. ACM.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. Harnessing deep neural networks with logic rules. In *Proc. of ACL*.

Matthew J. Johnson, David K. Duvenaud, Alex B. Wiltschko, Sandeep R. Datta, and Ryan P. Adams. 2016. Composing graphical models with neural networks for structured representations and fast inference. *Arxiv preprint arXiv:1603.06277*.

Theofanis Karaletsos, Serge Belongie, Cornell Tech, and Gunnar Rätsch. 2016. Bayesian representation learning with oracle constraints. In *Proc. of ICLR*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *Proc. of EMNLP*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Proc. of NIPS*, pages 1097–1105.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *AAAI*, pages 2267–2273.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2015. Molding cnns for text: non-linear, non-consecutive convolutions. In *Proc. of EMNLP*.

Jiwei Li, Dan Jurafsky, and Eudard Hovy. 2015. When are tree structures necessary for deep learning of representations?

Percy Liang, Hal Daumé III, and Dan Klein. 2008. Structure compilation: trading structure for features. In *Proc. of ICML*, pages 592–599. ACM.

Percy Liang, Michael I Jordan, and Dan Klein. 2009. Learning from measurements in exponential families. In *Proc. of ICML*, pages 641–648. ACM.

Shike Mei, Jun Zhu, and Jerry Zhu. 2014. Robust RegBayes: Selectively incorporating first-order logic domain knowledge into Bayesian models. In *Proc. of ICML*, pages 253–261.

Radford M Neal and Geoffrey E Hinton. 1998. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer.

Arvind Neelakantan, Quoc V Le, and Ilya Sutskever. 2016. Neural programmer: Inducing latent programs with gradient descent. In *Proc. of ICLR*.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. of EMNLP*, volume 1631, page 1642. Citeseer.

Jacob Steinhardt and Percy S Liang. 2015. Learning with relaxed supervision. In *Proc. of NIPS*, pages 2809–2817.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proc. of ACL*.

Bishan Yang and Claire Cardie. 2014. Context-aware learning for sentence-level sentiment analysis with posterior regularization. In *Proc. of ACL*, pages 325–335.

Wenpeng Yin and Hinrich Schutze. 2015. Multichannel variable-size convolution for sentence classification. *Proc. of CONLL*.

Han Zhao, Zhengdong Lu, and Pascal Poupart. 2015. Self-adaptive hierarchical sentence model. *arXiv preprint arXiv:1504.05070*.

Jun Zhu, Ning Chen, and Eric P Xing. 2014. Bayesian inference with posterior regularization and applications to infinite latent svms. *JMLR*, 15(1):1799–1847.