Deep Weighted MaxSAT for Aspect-based Opinion Extraction

Meixi Wu* **Wenya Wang** **Sinno Jialin Pan**
Nanyang Technological University, Singapore
MWU008@e.ntu.edu.sg, {wangwy, sinnopan}@ntu.edu.sg

Abstract

Though deep learning has achieved significant success in various NLP tasks, most deep learning models lack the capability of encoding explicit domain knowledge to model complex causal relationships among different types of variables. On the other hand, logic rules offer a compact expression to represent the causal relationships to guide the training process. Logic programs can be cast as a satisfiability problem which aims to find truth assignments to logic variables by maximizing the number of satisfiable clauses (MaxSAT). We adopt the MaxSAT semantics to model logic inference process and smoothly incorporate a weighted version of MaxSAT that connects deep neural networks and a graphical model in a joint framework. The joint model feeds deep learning outputs to a weighted MaxSAT layer to rectify the erroneous predictions and can be trained via end-to-end gradient descent. Our proposed model associates the benefits of high-level feature learning, knowledge reasoning, and structured learning with observable performance gain for the task of aspect-based opinion extraction.

1 Introduction

Aspect-based opinion extraction aims to identify opinion targets (or aspects) of a review corpus that indicate specific product features, as well as the opinion terms expressed towards the aspects. For example, in the sentence "*The wine list is excellent*", the aspect term is *wine list*, whereas the opinion term is *excellent*. Many deep learning models have been proposed for this task via enumerating high-level features (Liu et al., 2015; Xu et al., 2018a; Wang et al., 2017; Li and Lam, 2017; Yin et al., 2016; Wang et al., 2016). However, these methods fail to explicitly encode prior knowledge on the relationships among aspect terms and opinion terms which are crucial for the task at hand, as shown in earlier rule-based models (Hu and Liu, 2004; Qiu et al., 2011). As in the previous example, if *wine list* is extracted as an aspect term and it has dependency relation "nsubj" with *excellent* which is an objective, then we can deduce that *excellent* is an opinion term. Though in (Yu et al., 2019), rules are incorporated as constraints into a deep neural network, the constraints cannot be backpropagated to the feature learning process. Recently, Wang and Pan (2020) proposed a joint model to combine deep learning with logic rules via minimizing the discrepancy between them. Their approach, however, only indirectly guides deep learning in training without the ability to rectify the predictions according to logic rules in inference.

To address the aforementioned limitations for aspect-based opinion extraction, we propose a novel joint model **DeepWMaxSAT** to integrate logic knowledge via a weighted MaxSAT layer into a deep learning architecture. Specifically, DeepWMaxSAT consists of 1) a DNN layer that transforms an input embedding to a high-level feature representation; 2) a weighted MaxSAT layer that takes DNN outputs as the initial probabilistic evaluations on the logic variables and produces the values for the output logic variables corresponding to the head atoms of selected logic rules; 3) a conditional random field (CRF) (Lafferty et al., 2001) layer that generates structured outputs (label sequences) considering linear context interactions among the tokens in a sequence. Moreover, to fully inherit the advantages of both DNNs and logic programs, we adopt a form of residual connection that combines both DNN predictions and the outputs from the weighted MaxSAT layer with a learnable weight, which is then fed into the CRF layer.

It is worth noting that the weighted MaxSAT layer contains all the prior knowledge about the cor-

* This work was done when the first author was an undergraduate student with Nanyang Technological University.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 5618–5628,
November 16–20, 2020.

relations among aspect and opinion terms encoded in conjunctive normal form (CNF) for all the logic rules. For example, the association between the aspect term *wine list* and the opinion term *excellent* in the previous example can be expressed using CNF as $\neg\text{aspect}(list) \vee \neg\text{nsubj}(list, excellent) \vee \neg\text{obj}(excellent) \vee \text{opinion}(excellent)$, which is converted from the first-order-logic (FOL) rule $\text{obj}(excellent) \wedge \text{nsubj}(list, excellent) \wedge \text{aspect}(list) \Rightarrow \text{opinion}(excellent)$. A learnable weight is associated to each disjunctive clause in the CNF formula to indicate its confidence. The weighted MaxSAT layer is able to rectify DNN predictions according to preset rules, at the same time, the loss signal for the final predictions can be back-propagated smoothly through the weighted MaxSAT layer to DNN parameters to guide the training of the deep learning model. Though Wang et al. (2019) proposed a differentiable satisfiability solver that integrates MaxSAT into deep learning, they only assumed a fixed set of rules that are true in nature, making it less flexible for general NLP problems where data can be noisy. With this consideration, we adopt the attention mechanism to adaptively select useful rules in the weighted MaxSAT layer for each data instance and treat the learnable attention scores as rule weights. The intuition is that different data instances may fit to different rules with varying probabilities.

To summarize, our contributions include:

- We propose a novel attention-based weighted MaxSAT solver that can selectively rectify and update deep learning predictions according to the relevance of specific rules.
- An end-to-end joint model associating DNNs, logic reasoning and structured learning is introduced to enhance the model performance.
- We focus on evaluating the effectiveness of encoding manually-designed prior knowledge as logic rules into a deep architecture. To achieve that, a real NLP application, namely aspect-based opinion extraction is chosen which is noisy but contains certain syntactic regularities that are difficult to be captured by pure deep learning models.
- We demonstrate the generality of the proposed joint framework over different DNN systems and word embeddings on the task of aspect-based opinion extraction.

2 Related Work

Aspect-based Opinion Extraction Various deep learning approaches have been introduced for aspect-based opinion extraction, including context-based recurrent neural networks (Liu et al., 2015) and convolutional neural networks (Xu et al., 2018a), dependency-tree-based models (Yin et al., 2016; Wang et al., 2016), and attention-based models (Wang et al., 2017; Li and Lam, 2017). Despite the promising performaces, it is hard to interpret and explicitly encode prior knowledge for deep learning models. The prior knowledge has been commonly used in the earlier works by designing specific features and rules among aspect terms and opinion terms (Hu and Liu, 2004; Popescu and Etzioni, 2005; Wu et al., 2009; Qiu et al., 2011). Yu et al. (2019) used integer linear programming with explicit constraints for joint inference as a post-processing step. However, these rule-based methods fail to propagate training signal to the feature learning process, making them suboptimal. On the other dimension, graphical models were also proposed to model the contextual or syntactic interactions among the tokens (Jin and Ho, 2009; Li et al., 2010). However, the optimization process is usually non-trivial especially for complex graphical structures. Recently, Wang and Pan (2020) introduced a logic-informative deep learning model that converts the relations among aspect and opinion terms to logic rules. Nevertheless, the logic rules only implicitly guide the training process of DNN and fail to rectify DNN predictions directly.

Deep Learning with Logic Reasoning Recent years have witnessed an increasing focus on neural symbolic learning that combines deep learning systems with discrete symbolic rules (Garcez et al., 2012; Manhaeve et al., 2018; Dong et al., 2019; Sourek et al., 2018; Wang et al., 2019) by constructing a logic network or connecting the distributed systems with logic rules for reasoning and inference in the logic domains. Xu et al. (2018b) treated logic knowledge as semantic regularization in the loss function. For NLP applications, the neural-symbolic systems were recently proposed in (Rocktäschel et al., 2015; Guo et al., 2016) for relation and knowledge graph learning that embed logic into the same space as distributed features in a single system. Logical knowledge has also been incorporated as a form of posterior regulariza-

tion in (Hu et al., 2016) to enhance deep learning predictions. Moreover, logic rules can be used as evidences to construct adversarial sets (Minervini et al., 2017; Minervini and Riedel, 2018), or as a form of indirect supervision (Wang and Poon, 2018) to improve model training. Li and Srikumar (2019) further augmented deep learning models with logic neurons that can be trained together with the neural networks.

3 Problem Definition & Preliminary

3.1 Problem Definition

We treat the extraction problem as a sequence labeling task. Given a sequence of tokens $\{w_1, w_2, ...w_n\}$, sequence labeling produces a segmentation label y_i for each token w_i where $y_i \in Y = \{\text{B-ASP}, \text{I-ASP}, \text{B-OPN}, \text{I-OPN}, \text{O}\}$. We use BIO encoding scheme to differentiate whether the token is the beginning of an aspect/opinion term (B-ASP/B-OPN), inside an aspect/opinion term (I-ASP/I-OPN), or out of any targets (O).

A first-order-logic (FOL) rule or a clause has the form of $a_1 \wedge a_2 \wedge ... \wedge a_K \Rightarrow h$, where $a_1 \wedge a_2 \wedge ... \wedge a_K$ is the rule body containing a conjunction of atoms a_k, and h is the head atom. Here, an atom is an n-ary predicate $a_k = pred_k(x_1, ..., x_n)$ with $x_1, ..., x_n$ representing n variables. A ground atom assigns a constant to each varible in its argument. A set of FOL rules can be transformed to a conjunctive normal form (CNF) which is a conjunction of one or more disjunctive clauses, e.g., the clause $\neg a_1 \vee \neg a_2 \vee ... \vee \neg a_K \vee h$ is converted from $a_1 \wedge a_2 \wedge ... \wedge a_K \Rightarrow h$. Here, each disjunctive clause corresponds to an FOL rule. When the CNF formula is satisfised, all its corresponding FOL rules are true. In our setting, we treat the linguistic features, e.g., dependency relations, POS tags, and the segmentation labels as different predicates. For example, B-ASP(w_i) is a ground atom indicating w_i as the beginning of an aspect term. We utilise these atoms to form the CNF formula in the MaxSAT formulation.

3.2 Differentiable MaxSAT Solver

The maximum satisfiability problem (MAX-SAT) is the problem of determining the maximum number of satisfied clauses. Given a formula in CNF $c_1 \wedge ... \wedge c_m$ with m disjunctive clauses $c_1, ..., c_m$ over a total number of n different atoms $a_1, ..., a_n$, each atom takes one of the 2 assignments: $v_i \in \{-1, +1\}$ indicating its truth value. For each clause c_j, we denote its sign $\mathbf{s}_j$ corresponding to all the atoms by $\mathbf{s}_j = \{-1, 0, +1\}^n$, where $s_{ji} \in \mathbf{s}_j$ takes -1, 0 or $+1$ indicating the sign of atom a_i in clause c_j. 0 represents the absence of a_i. Then the MaxSAT problem can be casted into the following optimization problem:

$$\max_{v_i \in \{-1,1\}^n} \sum_{j=1}^{m} \bigvee_{i=1}^{n} \mathbf{1}\{s_{ji} v_i > 0\}. \quad (1)$$

To solve this problem, Wang et al. (2019) transformed (1) to the following objective by relaxing each discrete v_i to a continuous unit vector $\bar{\mathbf{v}}_i \in \mathbb{R}^k$ with respect to some "truth direction" $\mathbf{v}_\top$ through $P(v_i = 1) = \cos^{-1}\left(-\bar{\mathbf{v}}_i^T \mathbf{v}_\top\right)/\pi$.

$$\begin{aligned} \min_{\mathbf{V} \in \mathbb{R}^{d \times (n+1)}} \quad & \left\langle \mathbf{S}^\top \mathbf{S}, \mathbf{V}^\top \mathbf{V} \right\rangle \\ \text{s.t.} \quad & \|\bar{\mathbf{v}}_i\| = 1, i = \top, 1, \ldots, n, \end{aligned} \quad (2)$$

where $\mathbf{V} = [\mathbf{v}_\top, \bar{\mathbf{v}}_1, ...\bar{\mathbf{v}}_n] \in \mathbb{R}^{k \times (n+1)}$ and $\mathbf{S} = [\mathbf{s}_\top, \mathbf{s}_1, ..., \mathbf{s}_n]\,\mathrm{diag}(1/\sqrt{4\,|\mathbf{s}_j|}) \in \mathbb{R}^{m \times (n+1)}$. Here $\mathbf{s}_\top = \{-1\}^m$. The problem (2) can be solved via coordinate descent with the following update:

$$\bar{\mathbf{v}}_i = -\mathbf{g}_i / \|\mathbf{g}_i\|, \; \mathbf{g}_i = \mathbf{V}\mathbf{S}^\top \mathbf{s}_i - \|\mathbf{s}_i\|^2 \bar{\mathbf{v}}_i. \quad (3)$$

This update is guaranteed to converge to the global optimal as long as $k > \sqrt{2n}$. To obtain the final probabilistic evaluations for atom a_i, we convert the updated $\bar{\mathbf{v}}_i$ to $p(v_i = 1) = \cos^{-1}\left(-\bar{\mathbf{v}}_i^\top \mathbf{v}_\top\right)/\pi$.

4 Methodology

In this section, we present our proposed model in detail. To make the logic knowledge more effective that is able to directly rectify the erroneous predictions made by deep learning models, and at the same time adapt its rules selectively according to different data instances, we propose a neural-symbolic integration by incorporating an attention-based weighted MaxSAT layer. The attention mechanism is used to automatically select relevant logic rules according to each specific data instance and to weigh the importance of each rule that could affect the final objective. Furthermore, we also integrate a CRF layer to generate structured predictions. As a result, the joint framework inherit the advantage of high-level feature learning, knowledge reasoning and structured learning.

Figure 1 provides an overview of the proposed model. It consists of 3 layers: 1) a deep learning module that takes input embeddings $\mathbf{x}_1, ..., \mathbf{x}_N$ as

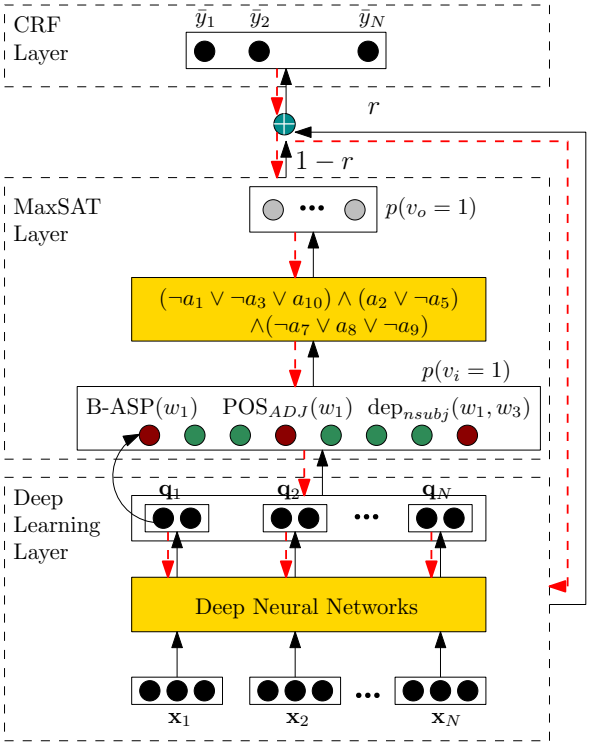


Figure 1: The proposed overall architecture.

inputs and generates a prediction for each word $\mathbf{q}_1, ..., \mathbf{q}_N$ via feature learning; 2) a weighted MaxSAT layer that takes deep learning predictions as the initial probabilistic evaluations $p(v_i = 1)$ of the input atoms a_i and generates probabilistic values $p(v_o = 1)$ of the output atoms; 3) a CRF layer that combines the outputs from the previous 2 layers with a residual connection to produce the final structured predictions $\bar{y}_1, ..., \bar{y}_N$. The joint model can be trained in an end-to-end manner via gradient descent, which is reflected with the dotted arrows in Figure 1. We illustrate each component in more detail in the sequel.

4.1 Deep Learning Layer

The deep learning layer aims to capture high-level feature representation for each word considering the complex interactions among different words within a setence[1]. We use a transformer model which takes a combination of word embedding $\mathbf{x}_i^e$ and POS tag embedding $\mathbf{x}_i^p$ as input and generates a hidden representation $\mathbf{h}_i$ for each word via a multi-layer self-attention mechanism. Specifically, at the l-th layer of the transformer, each attention head computes one interaction factor between each token and other tokens within the sentence in order to produce

$$\widetilde{\mathbf{h}}_{i,l}^c = \sum_{j=1}^{m} \alpha_{ij}^c (\mathbf{W}_v^c \widetilde{\mathbf{h}}_{j,i-1}), \quad (4)$$

$$\boldsymbol{\alpha}_i^c = \text{softmax}\left(\frac{(\mathbf{W}_q^c \widetilde{\mathbf{h}}_{i,l-1})(\mathbf{W}_k^c \mathbf{H}_{l-1})}{\sqrt{d}} \right), \quad (5)$$

where each $\mathbf{h}_{i,l-1}$ is a column vector of the matrix $\mathbf{H}_{l-1}$. $\{\mathbf{W}_v^c, \mathbf{W}_q^c, \mathbf{W}_k^c\}$ are the transformation matrices of the c-th attention head. Here we use C individual transformations. By integrating the C transformations, the resultant hidden vector is computed as $\widetilde{\mathbf{h}}_{i,l} = \mathbf{W}[\mathbf{h}_{i,l}^1 : \ldots : \mathbf{h}_{i,l}^C]$.

A Bi-GRU (gated recurrent unit) f_θ is then applied after the last layer of the transformer $\widetilde{\mathbf{h}}_{i,L}$ to produce context-sensitive hidden representations

$$\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i, \overrightarrow{\mathbf{h}}_i] = [f_\theta(\widetilde{\mathbf{h}}_{i,L}, \overrightarrow{\mathbf{h}}_{i-1}) : f_\theta(\widetilde{\mathbf{h}}_{i,L}, \overleftarrow{\mathbf{h}}_{i+1})].$$

The final prediction of each word is obtained via a fully-connected layer with a softmax activation function: $\mathbf{q}_i = \text{softmax}(\mathbf{W}_y l(\mathbf{h}_i) + \mathbf{b}_y)$, where

$$l(\mathbf{h}_i) = \tanh(\mathbf{W}_h[\mathbf{h}_i^\top : \mathbf{x}_{i-1}^l] + \mathbf{b}_h), \quad (6)$$

and $\mathbf{x}_{i-1}^l$ indicates the label embedding of the preceding token.

4.2 Weighted MaxSAT Layer

As discussed in Section 3.1, we convert FOL rules to CNF formulas which consist of multiple disjunctive clauses in order to be fed into the MaxSAT solver. In our problem setting, each atom in a clause is a 1-ary or 2-ary predicate, e.g., a clause in the form of $\neg\text{ASP}(Y) \vee \neg\text{POS}_{NOUN}(X) \vee \neg\text{dep}_{nsubj}(X, Y) \vee \neg\text{POS}_{ADJ}(X) \vee \text{OPN}(X)$ indicates that if Y is an aspect word with POS tag "NOUN", and Y has dependency relation "nsubj" with X, then we can deduce that X is an opinion word when it has POS tag "ADJ". This clause can be well fit into the following sentence "*The wine list is excellent*" for extracting *excellent* as an opinion word when *wine list* is correctly predicted as an aspect term. The clauses we adopt are shown and explained in Figure 2.

In the weighted MaxSAT layer, we define the set of all atoms $\{a_1, ..., a_n\}$ as the atoms appeared in Figure 2, including label atoms[2]

[1] It is flexible to adopt different deep learning models with various word embeddings. To demonstrate such flexibility, we use different DNNs and word embeddings in experiments. Here, we only describe a transformer-style DNN for illustration.

[2] ASP(Y) indicates Y is either labeled as B-ASP or I-ASP, similar for OPN(X).

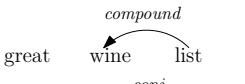
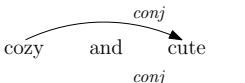
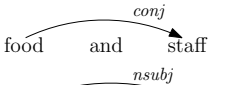
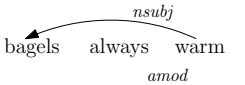
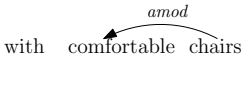
Clause	Example
$c_1 : \neg\text{ASP}(Y) \vee \neg\text{POS}_{NOUN}(Y) \vee \neg\text{dep}_{compound}(X,Y) \vee \neg\text{POS}_{NOUN}(X) \vee \text{I}-\text{ASP}(X)$	
$c_2 : \neg\text{OPN}(X) \vee \neg\text{POS}_{ADJ}(X) \vee \neg\text{dep}_{conj}(X,Y) \vee \neg\text{POS}_{ADJ}(Y) \vee \text{OPN}(Y)$	
$c_3 : \neg\text{ASP}(X) \vee \neg\text{POS}_{NOUN}(X) \vee \neg\text{dep}_{conj}(X,Y) \vee \neg\text{POS}_{NOUN}(Y) \vee \text{ASP}(Y)$	
$c_4 : \neg\text{ASP}(Y) \vee \neg\text{POS}_{NOUN}(Y) \vee \neg\text{dep}_{nsubj}(X,Y) \vee \neg\text{POS}_{ADJ}(X) \vee \text{OPN}(X)$	
$c_5 : \neg\text{OPN}(Y) \vee \neg\text{POS}_{ADJ}(Y) \vee \neg\text{dep}_{amod}(X,Y) \vee \neg\text{POS}_{NOUN}(X) \vee \text{ASP}(X)$	

Figure 2: Disjunctive clauses used in the weighted MaxSAT layer.

(e.g., ASP(Y), I-ASP(X), OPN(X)), POS atoms (e.g., POS$_{NOUN}(Y)$, POS$_{ADJ}(X)$) and dependency relation atoms (e.g., dep$_{compound}(X,Y)$, dep$_{conj}(X,Y)$, dep$_{nsubj}(X,Y)$, dep$_{amod}(X,Y)$). As shown in (2), the MaxSAT problem can be relaxed with the converted sign matrix $\mathbf{S}$ and atom value matrix $\mathbf{V}$. Here $\mathbf{S}$ is computed from the given clauses as our prior knowledge and kept fixed during training. To obtain $\mathbf{V} = [\mathbf{v}_\top, \bar{\mathbf{v}}_1, ...\bar{\mathbf{v}}_n]$, we take the softmax prediction from the deep learning layer as the initialized probabilistic value of each atom. Specifically, denote by $p(v_1 = 1), ..., p(v_n = 1) \in [0, 1]$ the probabilistic evaluations of all the atoms $a_1, ..., a_n$. If a_i is one of the label atoms, i.e., $a_i \in \{\text{ASP}(X), \text{I-ASP}(X), \text{OPN}(X)\}$, we take DNN predictions as the initial evaluations for the corresponding atoms, e.g., $p(v_i = 1) = \mathbf{q}_i^{\text{B-OPN}}$ when $a_i = \text{B-OPN}(X)$ and $\mathbf{q}_i^{\text{B-OPN}}$ is the DNN prediction for the class B-OPN. When a_i corresponds to the atom of POS tags or dependency relations, e.g., $a_i = \text{dep}_{nsubj}(X,Y)$, we use $0/1$ assignment for $p(v_i = 1)$ obtained through the Stanford Parser, where 0 indicates non-existence of the corresponding POS tag or dependency relation, and vice versa.

Different from existing works using a differentiable MaxSAT solver, we assign a probabilistic weight $w_j \in [0, 1]$ for each clause indicating its confidence of being true, which is updated during training. To adapt the logic knowledge into the noisy dataset, where each clause is not guaranteed to be always true for different data instances, we adopt an attention mechanism to compute the adaptive clause weight for each data instance, which measures the similarity between the DNN predictions and each specific clause grounding. Since in the real cases, each data instance may only satisfy at most 2 clauses, we use the sparsemax operator to transform the attention weights such that only 1 or 2 clauses are being chosen at each time. The procedure is shown as follows:

$$w_j^z = \text{sparsemax}(\mathbf{v}^{z\top} \hat{\mathbf{s}}_j), \tag{7}$$

where $\text{sparsemax}(\boldsymbol{\alpha}) = \underset{x \in \Delta^{N-1}}{\text{argmin}} \|\boldsymbol{x} - \boldsymbol{\alpha}\|^2$, and $\Delta^{N-1} = \{\boldsymbol{x} \in \mathbb{R}^N | \mathbf{1}^\top \boldsymbol{x} = 1, \boldsymbol{x} \geq \mathbf{0}\}$. Here w_j^z represents the weight for clause c_j corresponding to data instance z. $\mathbf{v}^z \in \mathbb{R}^{n-1}$ is the initial probabilistic evaluation vector for atoms $A = \{a_i\}_{i \neq n_h}$ except the head atom of the rule corresponding to data instance z. And $\hat{\mathbf{s}}_j = |\mathbf{s}'_j|$ where $\mathbf{s}'_j \in \mathbb{R}^{n-1}$ corresponds to the sign of each atom except the head atom of the rule. In our context, a data instance z corresponds to a pair of words (w_1, w_2) which are the instantiations for X and Y, respectively, in Figure 2. Intuitively, by using (7), the model tends to select the most relevant rules/clauses according to the similarity between the rule body and the vlaues of the associated groundings (e.g., POS tags, dependency relations and DNN predictions for each token).

With the incorporation of the attention-based weights of rules, the original MaxSAT objective can be transformed to the following form:

$$\begin{aligned} \min_{\mathbf{V} \in \mathbb{R}^{d \times (n+1)}} \quad & \left\langle \mathbf{U}^\top \mathbf{U}, \mathbf{V}^\top \mathbf{V} \right\rangle \\ \text{s.t.} \quad & \|\bar{\mathbf{v}}_i\| = 1, i = \top, 1, \ldots, n, \end{aligned} \tag{8}$$

where $\mathbf{V} = [\mathbf{v}_\top, \bar{\mathbf{v}}_1, ...\bar{\mathbf{v}}_n]$ and $\mathbf{U} = \mathbf{WS}$ with $\mathbf{S} = [\mathbf{s}_\top, \mathbf{s}_1, ..., \mathbf{s}_n] \, \text{diag}(1/\sqrt{4\,|\mathbf{s}_j|}) \in \mathbb{R}^{m \times (n+1)}$ and $\mathbf{W} = \text{diag}(w_j), j = 1, \ldots, m$. By using coordinate descent, the update for $\bar{\mathbf{v}}_i$ becomes

$$\bar{\mathbf{v}}_i = -\mathbf{g}_i / \|\mathbf{g}_i\|, \mathbf{g}_i = \mathbf{V}\mathbf{U}^\top \mathbf{u}_i - \|\mathbf{u}_i\|^2 \bar{\mathbf{v}}_i. \tag{9}$$

Note that we use (9) to compute $\bar{\mathbf{v}}_o$ until convergence with o being the index of the head atom

of the selected rules according to the attention mechanism. We then further convert the real vector to probabilistic evaluation via $p(v_o = 1) = \cos^{-1}\left(-\bar{\mathbf{v}}_o^\top \mathbf{v}_\top\right)/\pi$. For ease of illustration, for each data instance z, we denote by $p_o^z = p(v_o^z = 1) = f_{\text{MaxSAT}}(\{p(v_i^z)\}_{i\neq o})$ the output probability from the weighted MaxSAT layer. Intuitively, f_{MaxSAT} aims to produce a rule-satisfied evaluation to its corresponding head atom, given the DNN predictions of the input body atoms. When the DNN prediction for the head atom is not accurate, the MaxSAT layer is able to revise its value. In the meantime, the partial gradient of the final loss with respect to the MaxSAT output is backpropagated to the DNN parameters, making logic rules as a form of indirect supervision to the training of the DNN.

4.3 CRF Layer

To further mitigate the degradation problem caused by inaccurate MaxSAT updates or uncertain DNN predictions, we use a residual network with a trainable gate r to combine the outputs from both the DNN layer and the weighted MaxSAT layer as

$$\bar{\mathbf{y}}_i = r\mathbf{q}_i + (1-r)\mathbf{p}_i, \tag{10}$$

where $\mathbf{q}_i$ and $\mathbf{p}_i$ represent the outputs from the DNN and the MaxSAT layers, respectively.

On top of the combination, a CRF layer is performed to generate the structured prediction outputs, which takes into consideration of the sequential dependencies among entities. Denote by $\mathbf{x}$ and $\mathbf{y} = (y_1, \ldots, y_N)$ the input and the output of the CRF layer, respectively. The CRF layer computes conditional distributions as follows,

$$P(\mathbf{y}|\mathbf{x}) = \frac{\exp(f(\mathbf{x}, \mathbf{y}))}{\sum_{\mathbf{y}'} \exp(f(\mathbf{x}, \mathbf{y}'))}, \tag{11}$$

where $f(\mathbf{x}, \mathbf{y}) = \sum_i \log \psi_i(\mathbf{x}, \mathbf{y}) + \sum_{i'} \log \phi_{i'}(\mathbf{y})$. Here, $\psi_i(\mathbf{x}, \mathbf{y})$ and $\phi_{i'}(\mathbf{y})$ indicate the unary and pairwise potentials, respectively. To integrate the information from the preceding layers, we substitute $\psi_i(\mathbf{x}, \mathbf{y})$ with $\bar{\mathbf{y}}_i$ obtained via (10). The pairwise potential is determined via a trainable transition matrix specifying the score of transitioning from each label tag to other labels.

4.4 Training

The entire model can be trained in an end-to-end manner via gradient descent with the final loss function as

$$\mathcal{L} = -\frac{1}{D}\sum_{d=1}^{D} P(\hat{\mathbf{y}}_d|\mathbf{x}_d), \tag{12}$$

where $\hat{\mathbf{y}}_d$ is the ground-truth label sequence for data $\mathbf{x}_d$. During training, the objective updates the weighted MaxSAT layer according to (10) and (9) via:

$$\frac{\partial \mathcal{L}}{\partial \bar{\mathbf{v}}_o} = \frac{\partial \mathcal{L}}{\partial \bar{\mathbf{y}}} \frac{\partial \bar{\mathbf{y}}}{\partial p_o} \frac{\partial p_o}{\partial \bar{\mathbf{v}}_o}, \tag{13}$$

$$\frac{\partial \mathcal{L}}{\partial \bar{\mathbf{v}}_i} = \left(\frac{\partial \mathcal{L}}{\partial \bar{\mathbf{v}}_o}\right)^\top \frac{\partial \bar{\mathbf{v}}_o}{\partial \bar{\mathbf{v}}_i}, \tag{14}$$

$$\frac{\partial \mathcal{L}}{\partial w_j} = \left(\frac{\partial \mathcal{L}}{\partial \bar{\mathbf{v}}_o}\right)^\top \frac{\partial \bar{\mathbf{v}}_o}{\partial \mathbf{P}} \frac{\partial \mathbf{P}}{\partial w_j}, \tag{15}$$

where $\bar{\mathbf{v}}_o$ and $\bar{\mathbf{v}}_i$ represent the output index (head atom) and the input index (body atom), respectively. Following (Wang et al., 2019), we take the analytical form of the resulting gradients to compute (14) and (15), respectively.

Note that the gradients of DNN parameters (denoted by Θ) are obtained through backpropagating information from both the final loss function $\mathcal{L}$ and the MaxSAT gradient $\frac{\partial \mathcal{L}}{\partial \bar{\mathbf{v}}_i}$ via:

$$\frac{\partial \mathcal{L}}{\partial \Theta} = \left(\frac{\partial \mathcal{L}}{\partial \mathbf{q}} + \sum_i \left(\frac{\partial \mathcal{L}}{\partial \bar{\mathbf{v}}_i}\right)^\top \frac{\partial \bar{\mathbf{v}}_i}{\partial \mathbf{q}} + \sum_j \frac{\partial \mathcal{L}}{\partial w_j} \frac{\partial w_j}{\partial \mathbf{q}}\right) \frac{\partial \mathbf{q}}{\partial \Theta}. \tag{16}$$

5 Experiment

We conduct experiments on the benchmark dataset from SemEval Challenge 2014 task 4 (subtask 1) that consists of a restaurant domain and a laptop domain (Pontiki et al., 2014), and a restaurant corpus from SemEval 2016 task 5 (Pontiki et al., 2016). The details of each data are listed in Table 1. For preprocessing, we use NLTK toolkit for tokenization, POS tagging and generating dependency parse tree for each sentence. We use 1 GPU with model Tesla P100-PCIE-16GB to run our experiment. For the joint model, it takes around 20 minutes for an epoch with 3000 data instances and it takes 10 epochs to achieve the optimal performance.

Dataset	Description	Training	Test	Total
Restaurant14	SemEval-14 Restaurant	3,041	800	3,841
Laptop14	SemEval-14 Laptop	3,045	800	3,845
Restaurant16	SemEval-16 Restaurant	2,000	676	2,676

Table 1: Dataset description with number of sentences

5.1 Experimental Setting

Follow the setting in (Wang et al., 2016), the pre-training of word embedding is first conducted

Model		RNCRF	CMLA	Demb	GMTCMLA	DeepLogic	DeepWMaxSAT
Restaurant14	Aspect	84.93	85.29	84.24	84.50	85.24	**85.33**
	Opinion	84.11	83.18	-	85.20	84.37	**85.73**
Laptop14	Aspect	78.42	77.80	81.59	78.69	81.25	**81.33**
	Opinion	79.44	80.17	-	79.89	79.32	**80.34**
Restaurant16	Aspect	69.74	**75.21**	74.37	-	73.35	73.67
	Opinion	76.15	77.90	-	-	78.89	**79.67**

Table 2: Results on 3 benchmark datasets for aspect and opinion extraction.

using *word2vec* on Yelp Challenge dataset[3] and electronic dataset in Amazon reviews[4] for restaurant and laptop domain, respectively. Following (Vaswani et al., 2017), we add positional encoding on top of input representations in the transformer network. We assign 10 heads to the multi-head self-attention model, which generates attention weight parameters with dimension 10. We set the word embedding dimension as 300, POS-tag embedding as 50, hidden layer as 200, and label embedding as 25. For training, we adopt the adadelta optimizer with a learning rate of $2e^{-3}$ and a weight decay of $5e^{-4}$. All parameters are chosen based on cross-validation. To evaluate the model performance, F1 scores on non-negative classes are adopted, where the correctness of a prediction is fulfilled if and only if the predicted tag exactly matches the true label for each aspect/opinion term.

5.2 Overall Results

We evaluate our model performance by comparing with the following well-known baseline methods:

- RNCRF (Wang et al., 2016): A joint model combining a dependency-based recursive neural network with CRF to model syntactic interactions among aspect and opinion terms.
- CMLA (Wang et al., 2017): Coupled attention network with tensor-based interaction for co-extraction of aspect and opinion terms.
- Demb (Xu et al., 2018a): A convolutional neural network with domain-dependent and domain-independent word embeddings.
- GMTCMLA (Yu et al., 2019): Global inference with multi-task neural networks that regularize DNN predictions with integer linear programming.
- DeepLogic (Wang and Pan, 2020): Integrate deep learning with logic rules through minimizing a discrepancy loss.

The comparison results are shown in Table 2, and the last column corresponds to our proposed model. Clearly, our model achieves best performances on almost all the tasks across 3 datasets. The first 3 models represent pure deep learning methods by adopting either dependency trees (RNCRF), attention-based interactions (CMLA), or contextual interactions using convolutional neural network (Demb). These methods, however, only assume that the complex interactions among aspect terms and opinion terms can be captured via implicit feature learning. When feeding prior knowledge as constraints in integer linear programming, GMTCMLA is able to regularize deep learning predictions, but without the ability to backpropagate error information. Hence, its performance does not show clear improvement. DeepLogic is able to update the deep learning model by treating logic rules as indirect supervision. Without the capability to directly revise DNN outputs, it shows suboptimal performance compared to our proposed model.

We further conduct a qualitative analysis to demonstrate how the weighted MaxSAT (WMaxSAT) layer rectify the erroneous predictions made by deep neural networks. Some representative cases that WMaxSAT corrects DNN predictions are shown in Table 3. The left column shows predictions made by the deep learning model with the incorrectly predicted words marked in red. The right column shows the corresponding predictions made by applying a WMaxSAT layer on top of DNN outputs. It is clear that those mislabeling words are all corrected in this case, demonstrating the effect of our proposed model.

5.3 Ablation Analysis

To further demonstrate the effect of each component of our proposed model, we conduct ablation experiments with 6 different model set-

[3]http://www.yelp.com/dataset_challenge
[4]http://jmcauley.ucsd.edu/data/amazon/links.html

DNN prediction	WMaxSAT correction
[‘*pretentious*’ - **O**, ‘*and*’ - O, ‘*inappropriate*’ - B-OPN]	[“*pretentious*’ - **B-OPN**, ‘*and*’ - O, ‘inappropriate’ - B-OPN]
[‘*flan*’ - B-ASP, ‘*and*’ - O, ‘*sopaipillas*’ - **O**]	[‘*flan*’ - B-ASP, ‘*and*’ - O, ‘*sopaipillas*’ - **B-ASP**]
[‘*sauce*’ - B-ASP, ‘*cart*’ - **O**]	[‘*sauce*’ - B-ASP, ‘*cart*’ - **I-ASP**]
[‘*delivery*’ - B-ASP, ‘*times*’ - **O**]	[‘*delivery*’ - B-ASP, ‘*times*’ - **I-ASP**]
[‘*management*’ - B-ASP, ‘*accommodating*’ - **O**]	[‘*management*’ - B-ASP, ‘*accommodating*’ - **B-OPN**]

Table 3: Examples where the WMaxSAT layer corrects the DNN predictions.

Model Settings	Restaurant14		Laptop14		Restaurant16	
	ASP	OPN	ASP	OPN	ASP	OPN
DNN	84.59	84.71	79.21	77.88	72.28	80.87
DNN+CRF	84.71	85.67	**81.72**	79.41	72.45	81.15
DNN+WMaxSAT	85.47	85.26	81.41	78.84	73.41	**82.81**
DNN+WMaxSAT+CRF	**85.33**	**85.73**	81.33	**80.34**	**73.67**	79.67
DNN+MaxSAT+CRF	84.22	85.62	81.24	77.75	72.37	80.12
DNN+MaxSAT*+CRF	84.50	85.56	81.14	79.07	72.59	80.10

Table 4: Comparison with different model settings.

Model Settings	Restaurant14		Laptop14		Restaurant16	
	ASP	OPN	ASP	OPN	ASP	OPN
BERT	86.16	86.12	80.16	79.52	72.32	82.21
BERT+CRF	86.40	87.76	79.93	80.72	72.45	82.25
BERT+WMaxSAT	86.29	87.55	79.90	79.76	72.36	82.50
BERT+WMaxSAT+CRF	**86.71**	**88.01**	**80.54**	**81.02**	**73.60**	**82.59**
ELMO	85.28	84.88	72.76	78.19	71.33	81.44
ELMO+CRF	85.13	85.43	74.38	79.59	72.18	79.67
ELMO+WMaxSAT	**85.55**	85.57	**74.45**	79.77	72.19	**82.18**
ELMO+WMaxSAT+CRF	85.43	**85.70**	74.12	**79.91**	**72.65**	81.11

Table 5: Comparison with different model settings on BERT and ELMO pretrained word embeddings.

Clauses	c_1	c_2	c_3	c_4	c_5
ratio	0.22	0.14	0.17	0.30	0.08
Res14-ASP	85.59	85.18	85.55	85.00	85.18
Res14-OPN	85.12	85.71	85.29	85.52	85.66
Lap14-ASP	81.73	81.66	81.83	81.49	81.08
Lap14-OPN	79.38	79.58	79.24	79.49	79.46

Table 6: Utility rate and performance for each rule.

tings as shown in Table 4. The advantage of DNN+WMaxSAT over DNN alone in most cases reveals the power of using WMaxSAT to incorporate domain knowledge. Using CRF further improves the model performance through effective capturing of sequential correlations among terms. To show the advantage of using the proposed attention mechanism for rule weight computation, we compare with 2 other variations of the MaxSAT layer. DNN+MaxSAT+CRF assumes each logic rule as correct at all times (fixed weights to be 1.0). Whereas DNN+MaxSAT*+CRF assigns each rule with a unified weight which applies to all data instances. The rule weights in this model are randomly initialized and trained through the learning process. As can be seen, in most of the cases, attention-based WMaxSAT is most effective for aspect/opinion extraction.

Our proposed model is flexible to integrate any deep learning modules or pre-trained word embeddings. To show the generality and advantage of combining DNNs with logic reasoning and structured learning, we replace the transformer model in the deep learning layer with 2 other commonly used word embeddings, namely BERT (Devlin et al., 2019) and ELMO (Peters et al., 2018) followed by a BiGRU layer. The results for using different word embeddings with different model settings are shown in Table 5. Clearly, BERT achieves better performances than ELMO in general. It is worth noting that the weighted MaxSAT layer always brings performance gain when combined with the DNN model. The joint model over all the three components produces the best results when using BERT as the word embeddings. Whereas joining ELMO with WMaxSAT produces comparable performances with or without CRF.

To provide a clear idea of the effect for each logic rule described in Figure 2, we conduct experiments on feeding each single clause into the WMaxSAT layer as shown in Table 6. We observe the best performance on aspect extraction when only using c_1 for restaurant domain and c_3 for the laptop domain. For opinion extraction, c_2 is most effective for both domains. However, using separate rules are inferior than using all 5 rules for opinion extraction. We also analyze the percentage of each rule being selected during training, as shown in the second row of Table 6. On average, most rules has about 20% chance of being selected, which shows that the attention model is able to select diverse rules according to different data characteristics.

In the previous experiments, we initialize the residual connection gate r as 1.0 and update it through the training process. To demonstrate the effect of different initializations for this hyper-parameter, we conduct another experiment on varying the value of r from 0.1 to 1.0. As shown in Figure 3, the f1 scores do not fluctuate substan-

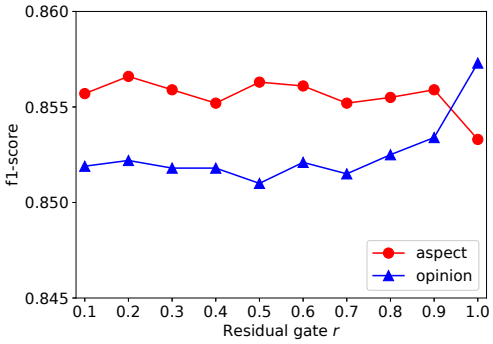


Figure 3: Sensitivity study for residual gate r on restaurant-14 dataset.

tially when $0.1 \leq r \leq 0.9$. When $r = 1.0$, there is a clear change of f1 scores. The reason might come from the fact that some logic rules are not always feasible for the actual noisy dataset, especially when some general objects which should be regarded as aspect terms according to the rules are not labeled as aspect terms. For example, in the sentence "*This place is amazing*", *amazing* is labeled as an opinion term whereas *place* is not labeled as an aspect term, which contradicts with rule c_4. When training with $r < 1.0$, the combination of label supervision and rule c_4 may result in missing the opinion term *amazing* given *place* is not an aspect term. In other words, the joint model tries to find a tradeoff between the labels and the rules that makes the result of aspect extraction and opinion extraction more balanced, instead of the evident performance difference when $r = 1.0$.

6 Conclusion

We propose a novel joint model that inherits the advantage of high-level feature learning, logic reasoning and structured learning which can be trained smoothly in an end-to-end manner. To adapt logic knowledge with noisy real applications, we introduce an attention mechanism to generate an adaptive weight corresponding to each data instance for each logic rule. The attention weights control the information flow between deep neural networks and the MaxSAT layer which automatically weigh the relevance of each rule towards the data given. Extensive experiments are conducted to verify both quantitatively and qualitatively the effectiveness of the proposed model.

Acknowledgement

This work is supported by NTU Nanyang Assistant Professorship (NAP) grant M4081532.020, 2020 Microsoft Research Asia collaborative research grant, and Singapore Lee Kuan Yew Postdoctoral Fellowship.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Honghua Dong, Jiayuan Mao, Tian Lin, Chong Wang, Lihong Li, and Denny Zhou. 2019. Neural logic machines. In *ICLR*.

Artur S. d'Avila Garcez, Krysia B. Broda, and Dov M. Gabbay. 2012. *Neural-Symbolic Learning System: Foundations and Applications*. Springer Science & Business Media.

Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2016. Jointly embedding knowledge graphs and logical rules. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 192–202, Austin, Texas. Association for Computational Linguistics.

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

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2410–2420, Berlin, Germany. Association for Computational Linguistics.

Wei Jin and Hung Hay Ho. 2009. A novel lexicalized hmm-based learning framework for web opinion mining. In *ICML*, pages 465–472.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289.

Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Ying-Ju Xia, Shu Zhang, and Hao Yu. 2010. Structure-aware review mining and summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 653–661, Beijing, China. Coling 2010 Organizing Committee.

Tao Li and Vivek Srikumar. 2019. Augmenting neural networks with first-order logic. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 292–302, Florence, Italy. Association for Computational Linguistics.

Xin Li and Wai Lam. 2017. Deep multi-task learning for aspect term extraction with memory interaction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2886–2892, Copenhagen, Denmark. Association for Computational Linguistics.

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1433–1443, Lisbon, Portugal. Association for Computational Linguistics.

Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. 2018. Deepproblog: Neural probabilistic logic programming. In *NeurIPS*, pages 3749–3759.

Pasquale Minervini, Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. 2017. Adversarial sets for regularised neural link predictors. In *UAI*.

Pasquale Minervini and Sebastian Riedel. 2018. Adversarially regularising neural NLI models to integrate logical background knowledge. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 65–74, Brussels, Belgium. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 339–346, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational Linguistics*, 37(1):9–27.

Tim Rocktäschel, Sameer Singh, and Sebastian Riedel. 2015. Injecting logical background knowledge into embeddings for relation extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1119–1129, Denver, Colorado. Association for Computational Linguistics.

Gustav Sourek, Vojtech Aschenbrenner, Filip Zelezný, Steven Schockaert, and Ondrej Kuzelka. 2018. Lifted relational neural networks. *JAIR*, 62.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Hai Wang and Hoifung Poon. 2018. Deep probabilistic logic: A unifying framework for indirect supervision. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1891–1902, Brussels, Belgium. Association for Computational Linguistics.

Po-Wei Wang, Priya L. Donti, Bryan Wilder, and J. Zico Kolter. 2019. Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. In *ICML*, volume 97, pages 6545–6554.

Wenya Wang and Sinno Jialin Pan. 2020. Integrating deep learning with logic fusion for information extraction. In *AAAI*.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. Recursive neural conditional random fields for aspect-based sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 616–626, Austin, Texas. Association for Computational Linguistics.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2017. Coupled multi-layer tensor network for co-extraction of aspect and opinion terms. In *AAAI*, pages 3316–3322.

Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. 2009. Phrase dependency parsing for opinion mining. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1541, Singapore. Association for Computational Linguistics.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2018a. Double embeddings and CNN-based sequence labeling for aspect extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 592–598, Melbourne, Australia. Association for Computational Linguistics.

Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Van den Broeck. 2018b. A semantic loss function for deep learning with symbolic knowledge. In *ICML*, pages 5502–5511.

Yichun Yin, Furu Wei, Li Dong, Kaimeng Xu, Ming Zhang, and Ming Zhou. 2016. Unsupervised word and dependency path embeddings for aspect term extraction. In *IJCAI*, pages 2979–2985.

Jianfei Yu, Jing Jiang, and Rui Xia. 2019. Global inference for aspect and opinion terms co-extraction based on multi-task neural networks. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 27(1):168–177.