Empirical Study of (Probabilistic) Soft Logic on Neural Network for Part-of-Speech Tagging and Sentiment Classification

Mourad Gridach

High Institute of Technology
Ibn Zohr University, Agadir, Morocco
m.gridach@uiz.ac.ma

Abstract Deep neural networks have emerged as a flexible framework that achieved state-of-the-art performance in many NLP applications such as machine translation, named entity recognition, sentiment analysis and part-of-speech tagging. The main advantage of these neural models is their ability to learn useful representations without hand-engineering features. While this success, these models still suffer from the interpretability issue. More recently, probabilistic soft logic (PSL) is a promising framework based on first-order logic that achieves interesting results in both computer vision and NLP by capturing semantic relationships between entities. Moreover, unifying knowledge driven modeling approaches and data-driven approaches is a promising framework that will have an exciting impact on structured based problems. In this paper, we developed NeuralGLogic a generalization framework of the previous model proposed by [1] that combines deep neural networks with logic rules built either using Soft Logic (SL) or Probabilistic Soft Logic (PSL). Furthermore, we evaluate our framework on different neural networks architectures applied to two NLP tasks: sentiment classification and part-of-speech tagging. Experimental results showed that we were able to improve the results over the baselines and outperformed all the previous state-of-the-art systems emphasizing the utility of both SL and PSL rules in reducing the uninterpretability of the neural models thus validating our intuition.

1 Introduction

Deep neural networks are powerful machine learning models able to learn interesting representations from data. In recent years, they achieved state-of-the-art results in various domains and difficult problems such as speech recognition [2,3], computer vision [4] and computer games [5]. In natural language processing, much of the work with neural nets has tackled neural machine translation [6,7], language modeling [8], named entity recognition [9,10,11,12], churn prediction [13,14] and sentiment analysis [15,16]. In addition, these models can take advantage from backpropagation algorithm [17] for training.

Despite the success of these neural models in learning useful representations, they still suffer from some problems. To learn useful representations, neural models rely heavily on massive datasets that cause them to learn uninterpretable and sometimes counter-intuitive features [18]. Furthermore, it is hard to incorporate human intention

allowing these models to capture the intended features since the cognitive process of humans learn from two different sources: samples (same as neural models) and other general structured knowledge [19]. Recent work in the computer vision community tackled this problem by modifying the architecture of these models in order to reduce their interpretability [20].

To tackle the problem of uninterpretability of the neural models, we propose to use the powerful expressiveness of both soft logic and probabilistic soft logic [21], which has been proven to be very useful in capturing human intention in natural language processing [1], computer vision applications such as Semantic Image Interpretation [22], recommender systems [23] and reasoning systems such as causal discovery [24]. They are useful frameworks for reasoning in relational domains where the logical atoms are used to represent the random variables, and first-order logic rules are used to capture dependencies between these random variables [25]. The main difference between SL/PSL and classical logic is allowing soft truth values in the interval $[0, 1]$ instead of just two values 0 or 1.

In this paper, we present NeuralGLogic a novel and general framework that combines deep neural networks with either SL rules or PSL rules. Our framework is a generalization of the system developed by [1] in four ways:

1. We applied their technique to one more NLP task namely Part-of-Speech tagging, and we were able to improve the performance over the baseline.
2. We add more PSL rules to the sentiment classification task while they used one logic rule.
3. We applied the framework to different neural networks architectures for validation.
4. While they use soft logic, we demonstrate the effectiveness of using probabilistic soft logic for sentiment classification to represent structural relations between segments in the text.

We apply the framework to two NLP tasks: sentiment classification and part-of-speech tagging where we took advantage of the hierarchical structure of the text to construct relational features that helped the system in the prediction task. For sentiment classification, we used two types of relations: *Contrast Relation* and *Neighborhood Relation*. For the part-of-speech (POS) tagging task, we exploit the fact that each sentence should have at least one verb and one noun where we want our model to capture this constraint.

For sentiment classification task, we evaluate our approach on four benchmarks. For POS tagging, we use the Wall Street Journal (WSJ) portion of Penn Treebank (PTB) for evaluating our model. Experimental results showed that we were able to improve the performances over the baselines on all benchmarks for sentiment classification and WSJ benchmark for POS tagging. The results confirm our intuition about the effectiveness of unifying neural models and SL/PSL in one framework.

2 Background

2.1 Posterior Regularization and Knowledge Distillation

[26] proposed posterior regularization, where they use constraints on posterior distributions of structured latent-variable models which allowed them to include indirect

(or weakly) supervised learning. They applied the framework to various applications such as multi-view learning, cross-lingual dependency grammar induction, unsupervised part-of-speech induction, and bitext word alignment. The main idea is to penalize the log-likelihood of a specific model with the KL divergence between the desired distribution that includes prior knowledge and the model posteriors.

The following equation defines the posterior regularized likelihood:

$$G(\theta, q) = \lambda_1 \mathcal{L}(\theta) - \lambda_2 \mathcal{M}(\theta) \tag{1}$$

where $\mathcal{M}(\theta)$ has the following form:

$$\mathcal{M}(\theta) = \min_{q \in \mathcal{Q}} KL(q(Y)||p_\theta(Y|X)) \tag{2}$$

The two hyperparameters λ_1 and λ_2 are used to balance the choice between the likelihood and the posterior regularization. Q is a set of valid distributions representing the constrained posteriors. It is defined as the following:

$$Q = \{q(Y) : \mathbb{E}_q[\phi(X, Y) \leq b\} \tag{3}$$

where $\phi(x, y)$ represents the constraint features and b is the bound of constraint feature expectations. Therefore, we define the constrained posteriors Q regarding constraints features and their expectations. During the learning process, they directly enforced decomposable regularization on the next moments of latent variables, which allowed them to maintain the computational efficiency of the unconstrained model while guaranteeing desired constraints hold in expectation.

Knowledge distillation is a concept introduced by [27] and it is one of the successful frameworks where the main idea is using a simple machine learning model to learn a complex task by imitating the solution of another flexible model, mostly a large ensemble of models. The first simple model called the student while the second is called the teacher model. In general, we can interpret this concept as a transfer of knowledge learned by the teacher model, usually more expensive to train, as "soft target" labels for training the student model. Knowledge distillation framework was widely used in many applications in natural language processing [28]), computer vision [29,30] and recommendation systems [31].

2.2 Sentiment Analysis

Sentiment analysis, also known as opinion mining, is a Natural Language Processing (NLP) task that receives much attention these years where the main goal is to identify the sentiment polarity of a sentence to sentiment classes such as positive or negative, or more fine-grained classes such as very positive, positive, neutral. In the last decade, sentiment analysis systems play an important role in helping the development of many online applications for customer reviews and public opinion analysis. More general,

traditional sentiment analysis systems focus on classical opinion such as binary classification (positive or negative), while others developed systems for multiple categories such as six basic emotions (anger, happiness, fear, sadness, disgust, and surprise). Sentiment systems can then be used to identify sentiment categories from sentences.

2.3 Part-of-Speech Tagging

Part-of-Speech (POS) tagging, also called word-category disambiguation or grammatical tagging, is the linguistic sequence labeling task. It was the early first stages of deep language understanding, and its importance has been well recognized in the natural language processing community and widely tackled NLP applications. Given a sentence, the goal of POS tagging is to label each word with a unique tag that indicates its syntactic role. It could be a plural noun, adverb, verb, adjective, preposition.

Natural language processing (NLP) systems, like syntactic parsing [32,33], entity coreference resolution [34], information retrieval [35], word sense disambiguation and text-to-speech [36] are becoming more robust, in part because of utilizing output information of POS tagging systems.

3 Our approach

In this section, we highlight SL and PSL used in our model. We note that throughout this section, we use logic rules to denote both the soft logic rules and probabilistic soft logic rules. Figure 1 illustrates our model architecture.

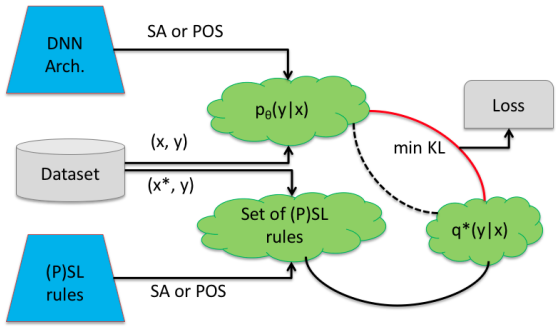

Figure 1: Our model architecture.

3.1 Soft Logic and Probabilistic Soft Logic

Soft Logic for POS. It is a modified first-order logic framework [21]. While classical logic values can take two values "0" or "1", soft logic enables continuous truth values from the interval $[0, 1]$ allowing more flexibility for encoding values. For POS

tagging, given a sentence S containing n words where $S = \{x_1, ..., x_n\}$, we define $T_S = \{t_1, ..., t_n\}$ to be the set of possible tags of the sentence S. We expect that each sentence contain at least one verb and one noun. We capture these two constraints using two soft logic rules:

$$Sentence(S) \implies \neg f_v(S, Ts) < 1$$

$$Sentence(S) \implies \neg f_n(S, Ts) < 1$$

In the first rule, we define $f_v(S, T_S)$ to be the number of verbs in the sentence S which we expect not to be less than 1. We use the same setting to define $f_n(S, T_S)$ as the number of nouns in a sentence.

Probabilistic Soft Logic for Sentiment Classification. It is a Statistical Relational Learning (SRL) method that allows doing probabilistic reasoning in relational domains. As an SRL framework, PSL uses soft logic to specify rules in order to capture the structure and relations in a specific domain. We represent a sentence S as a composition of two segments S_1 and S_2 related by a keyword, which belongs to either a contrast relation or neighborhood relation. We define two PSL rules to capture different relations between segments in a sentence:

- *Contrast Relation:* if a contrast relation relates two segments S_1 and S_2, then if S_1 has negative sentiment (NegS), then S_2 will have a positive sentiment (PosS) which will be the same for the whole sentence S. Also, if S_1 has positive sentiment, then both S_2 and S will have a negative sentiment. Therefore, we can derive four PSL rules:

$$S(S_1,S_2) \wedge Cont(S_1,S_2) \wedge NegS(S_2) \Rightarrow NegS(S)$$

$$S(S_1,S_2) \wedge Cont(S_1,S_2) \wedge PosS(S_2) \Rightarrow PosS(S)$$

$$S(S_1,S_2) \wedge Cont(S_1,S_2) \wedge NegS(S_1) \Rightarrow PosS(S)$$

$$S(S_1,S_2) \wedge Cont(S_1,S_2) \wedge PosS(S_1) \Rightarrow NegS(S)$$

In order to capture a contrast relation, we consider all the sentences containing the keywords “but”, “despite”, “though”, “although”, “while” and “however”. To illustrate this relation, consider the following sentence S: *“somewhat blurred, but kinnear’s performance is razor sharp”*. It contains two segments S_1: *“somewhat blurred”* and S_2: *“kinnear’s performance is razor sharp”*related by the keyword *“but”*. The segment S_2 has a negative sentiment while the second carries a positive sentiment, thus the sentiment of the whole sentence S will be positive. We found that 17% and 20% of sentences respectively from the SST2 and MR datasets contain a contrast keyword.

- *Neighborhood relation:* given two segments S_1 and S_2 in a sentence S where S_1 comes before S_2, if segment S_1 has positive sentiment, then segment S_2 maintains the same positive sentiment as well as the whole sentence S. The same procedure can be applied if S_1 has negative sentiment. We can derive four PSL rules like the following:

$$S(S_1,S_2) \wedge Neigh(S_1,S_2) \wedge NegS(S_2) \Rightarrow NegS(S)$$

$$S(S_1,S_2) \wedge Neigh(S_1,S_2) \wedge PosS(S_2) \Rightarrow PosS(S)$$

$$S(S_1,S_2) \wedge Neigh(S_1,S_2) \wedge NegS(S_1) \Rightarrow NegS(S)$$

$$S(S_1,S_2) \wedge Neigh(S_1,S_2) \wedge PosS(S_1) \Rightarrow PosS(S)$$

In this paper, we use the keyword *"and"* to express the conjunction between segments in a sentence. Let us consider the sentence S: *"this clever caper movie has twists worthy of David Mamet and is enormous fun for thinking audiences"* where it has two segments S_1: *"this clever caper movie has twists worthy of David mamet"* and S_2: *"is enormous fun for thinking audiences"*. The segment S_1 carries a positive sentiment and also the segment S_2 keeps carrying the same sentiment, which means that the sentiment of the whole sentence will be positive. We found that around 30% of sentences in both datasets contain a neighborhood keyword. We note that the strategy we followed for this relation is to consider just sentences containing the keyword "and" where we eliminate those containing a contrast keyword since we consider that contrast relation is much stronger than the neighboorhood relation.

In addition to the two previous relations, we can add two more relations: a *negation relation* where a sentence contains a negation word such as "cannot", "never", "hardly", "nothing", "neither", etc., in this case, the segment's sentiment is flipped. The fourth relation is called the *non-contrast relation* where a contrast relation does not relate the segments, so if S_1 and S_2 are non-contrast segments, and S_1 carries a positive sentiment, then S_2 also has a positive sentiment. We leave these two relations for future work.

3.2 NeuralGLogic: Unifying Deep Networks and Logic Rules

We define the set of PSL and SL rules in the form of functions $f_r \in \mathcal{X}\mathcal{Y} \rightarrow \mathbb{R}_+$ where r is the index of a specific logic rule, $\mathcal{X}$ is the space of inputs and $\mathcal{Y}$ is the space of outputs. For each rule, we will assign non negative weight $W_r \in \mathbb{R}_+$ (see Equation (4) and Equation(5)). The goal of these weights is to specify how an assignment will be penalized if a rule is not satisfied, thus measuring the importance of each rule [37]. There are two ways to define the weights: learn them or define their values if we are dealing with prior domain knowledge. We choose the second way since we already know our logic rules and the applied domain.

To unify the neural models with logic rules, we employ the posterior regularization (PR) concept of [26]. NeuralGLogic uses the PR framework by adding the logic rules as a regularization term to the neural networks. On the one hand, our goal is to guide the model towards desired behavior by using the logic rules considered as constraints functions. We enforce these logic rules regarding expectation, and we claimed that each logic rule $f_r(x, y)$ is true, which we formulate as an expectation term (the second term in Eq. (4)). On the other hand, we want that the prediction of both the neural network and logic rules to be close. We use the KL-divergence to measure this closeness (the first term in Eq. (4)). To satisfy these constraints, one can solve the following optimization problem:

$$\min_{q \in \mathcal{S}} KL(q(Y)||p_\theta(Y|X)) - C \sum_r W_r \, \mathbb{E}_q[f_r(X,Y)] \tag{4}$$

where S denotes the appropriate distribution space, $p_\theta(Y|X)$ is the conditional probability defined by the neural network and C is the regularization parameter. It should be noted that problem (4) is convex and has a closed-form solution given by the following (complete proof is given in [1]):

$$q^*(Y) \propto p_\theta(Y|X) \;\; \exp\{C \sum_r W_r f_r(X,Y)\} \tag{5}$$

We use the distillation objective function (Equation 6) developed by [27] for training our model. NeuralGLogic can also be seen as a transfer of knowledge from the logic rules to the CNN:

$$\theta^{(t+1)} = arg \min_{\theta \in \Theta} \; \frac{1}{m} \sum_{i=1}^{m} (1-\lambda) l(y_i, \sigma_\theta(x_i)) + \lambda l(s_i^{(t)}, \sigma_\theta(x_i)) \tag{6}$$

where l denotes the cross entropy loss function; m is the training size; $\sigma_\theta(x)$ is the softmax output of p_θ on x, $s_i^{(t)}$ is the soft prediction vector of $q^*(Y)$ on x_i at iteration t and the imitation parameter $\lambda \in [0,1]$ balances the importance between imitating the soft predictions $s_i^{(t)}$ and predicting the true hard labels y_i.

4 Experimental Results and Discussion

We evaluate our model on various sentiment classification and POS tagging benchmarks for English to demonstrate its effectiveness compared to extensive other state-of-the-art models. In order to have accurate results, we choose to use four different neural networks architectures within the same task (sentiment classification) and adding POS tagging task to show the generalization of the method on another task.

4.1 Sentiment Classification

In the sentiment classification task, we select four previous models based on deep neural networks in order to validate our framework. For each model, we develop three variant models using NeuralGLogic: we call the first model "NeuralGLogic-contrast" which combines a given neural network architecture with PSL rules implementing the contrast relation. Next, we call the second model "NeuralGLogic-neighb" which combines the same neural architecture with PSL rules implementing the neighborhood relation. The third model "NeuralGLogic-NC" uses the contrast and neighborhood relations.

For the hyperparameters, we use $\lambda(t) = 1 - 0.85t$ for the imitation parameter and we set the regularization parameter to *C = 300*. All the previous values are selected based

on the performance of the model on Stanford Sentiment Treebank (SST2) [38] dev set. We evaluate our models on five benchmarks namely Stanford Sentiment Treebank (SST1) - an extension of MR, SST2 [38], Subjectivity dataset (Subj) (Pang and Lee, 2004), and Movie Review (MR) [39]. For this dataset, we used 10-fold cross-validation.

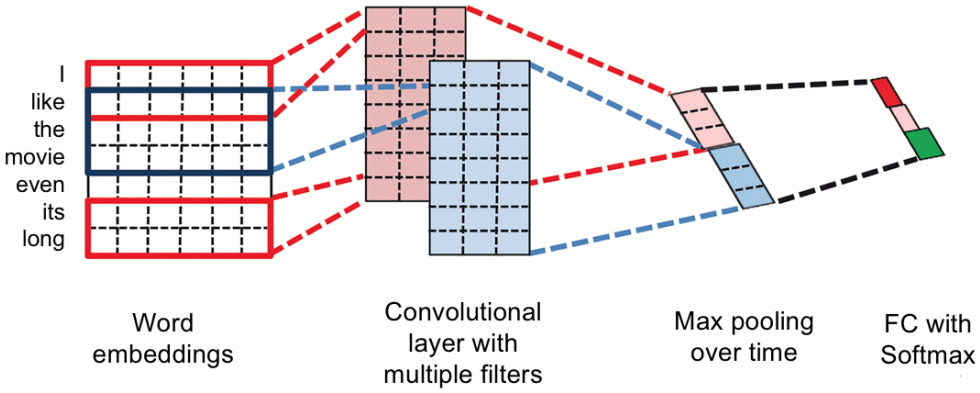

Figure 2: The CNN architecture developed by [15]

In the first architecture, we use the neural network developed by [15], which uses convolutional neural networks with pretrained word embedding. Figure 2 shows the architecture of this model. We use the CNN non-static version proposed by the author, and we follow the same hyperparameters. Table 2 depicts the experimental results. They represent the performance of NeuralGLogic applied to the three types of PSL relations in the sentiment classification task: the contrast relation, the neighborhood relation and a combination of the two relations. On both datasets, the second model using the contrast relation performs better than the model using neighborhood relation. While we do not have a complete explanation to these results, we believe that it is caused by the fact that the presence of a contrast keyword can have a significant effect on the sentence polarity while for the neighborhood keyword “and” that we used can sometimes have a neutral effect on the polarity sentence. As we expect, the “NeuralGLogic-NC” model combining the two relations give us the best performance. We compare the result of our models with other previous competitive systems. Table 1 presents the experiments. “NeuralGLogic-NC” outperformed the MVCNN model respectively by +0.5 points and the CNN-rule-q model by +0.6 points in accuracy.

In our second experiments, we consider the model developed by [40]. They use convolutional neural networks, and the goal is to make the system more adapt for two transformations: transformable convolution and transformable pooling, which allowed them to handle more complex features. Another advantage of this model is its ability to be integrated by other models in order to generate new transformable networks. In their paper, they developed two models namely, Transformable Dynamic convolutional neural network (TF-DCNN) and Transformable Multichannel CNN (TF-MCCNN). The results showed that TF-MCCNN obtained better results than TF-DCNN on Stanford

Model	SST2	MR
NeuralGLogic-neighb	89.1	81.8
NeuralGLogic-contrast	89.5	82.1
NeuralGLogic-NC	**89.9**	**82.4**
CNN [15]	87.2	81.3
CNN-rule-q [1]	89.3	81.7

Table 1: Classification accuracy (%) on two benchmarks: SST2 and MR of our system using [15] system as the baseline. Comparison of our system against [1] system with a single logic rule.

Model	SST1	SST2	Subj	TREC
NeuralGLogic-neighb	**49.4**	89.2	94.1	93.7
NeuralGLogic-contrast	49	89.4	94.2	93.6
NeuralGLogic-NC	49.2	**89.8**	**94.2**	**93.8**
TF-MCNN	49.1	88.5	94.2	93.5

Table 2: Classification accuracy (%) of our NeuralGLogic-based system using TF-MCNN as baseline on four benchmarks: SST1, SST2, Subj and TREC.

Sentiment Treebank (SST2) [38]. In our experiments, we used the SST2 dataset and the TF-MCCNN as our baseline model.

The results in Table 2 shows that adding the two types of logic rules to the baseline model (TF-MCNN) improved the performances. $NeuralGLogic - neighb$ performed better than than $NeuralGLogic-contrast$ model on SST1 dataset while on the other three datasets (SST2, Subj and TREC) $NeuralGLogic-contrast$ outperformed $NeuralGLogic - neighb$. By combining the two logic rules ($NeuralGLogic - NC$, we were able to improve the performances on the three datasets (SST2, Subj, and TREC), while on SST1 dataset the $NeuralGLogic - neighb$ still performed better. The main reason why we were not able to improve the performances because mainly the $NeuralGLogic-contrast$ model did not boost the results over the baseline model (TF-MCNN).

The third model employed in our experiments is multi-group norm constraint CNN (MGNC-CNN) [41]. The model is based on CNN architecture and uses multiple sets of word embeddings for the sentiment classification task. The main idea behind MGNC-CNN is extracting features from all the pretrained word embeddings, which could be with different dimensionality, independently and concatenate them into one single feature vector. In the experiments, we use Stanford Sentiment Treebank (SST2) [38] to evaluate our framework. Table 2 depicts the experiments, where it is clear that the $NeuralGLogic - NC$ outperformed all the models on the three datasets (SST2, Subj, and TREC) while $NeuralGLogic - neighb$ performed better on SST1 dataset.

In the last experiments, we consider the model developed by [42]. They also use a CNN-based architecture. The model has a 3D CNN structure featured by spatial pyramid pooling (SPP) brought from the object detection in computer vision. By combin-

Model	SST1	SST2	Subj	TREC
NeuralGLogic-neighb	**49.27**	88.87	94.77	95.87
NeuralGLogic-contrast	48.83	89.03	94.92	95.59
NeuralGLogic-NC	49.21	**89.31**	**95.05**	**95.93**
MGNC-CNN	48.65	88.35	94.11	95.52

Table 3: Results of our system using MGNC-CNN as baseline.

Model	SST1	SST2	Subj	TREC
NeuralGLogic-neighb	**51.6**	89.9	94.8	95.8
NeuralGLogic-contrast	51.1	90	95.1	96.1
NeuralGLogic-NC	51.5	**90.3**	**95.3**	**96.2**
3D-SPP CNN	50.8	89.5	94.5	95.8

Table 4: Classification accuracy (%) of our system using 3D-SPP CNN model as baseline on four benchmarks: SST1, SST2, Subj and TREC.

ing 3D convolutions and SPP, their model was able to capture more complex inner-structure in sentences. Moreover, SPP allows the model to handle the sentence length variety issue by splitting sentences into various length portions for pooling operation. In the experimental settings, we validated our models based on [42] system using Stanford Sentiment Treebank (SST2). Table 2 shows the experimental results. We can see that we get similar performances compared to the two previous models where $NeuralGLogic{-}neighb$ performed better on SST1 dataset, and $NeuralGLogic{-}NC$ outperformed all the models on the three datasets (SST2, Subj, and TREC).

4.2 POS tagging

The WSJ dataset contains 45 different POS tags. We follow the same standard split where we took section 0-18 as training data, section 19-21 as development data and lastly section 22–24 as test data. For the neural network hyperparameters, we followed [9]. Based on the results obtained on the dev set, we set $\lambda(t) = 1 - 0.85t$ for the imitation parameter and *C = 300* for the regularization parameter.

We use the end-to-end neural network developed by [9] as our base network for the POS tagging task. For the experiments, we use the Wall Street Journal (WSJ) portion of the Penn Treebank (PTB). We developed three different models: "DeepSLV" model use the logic rule based on the number of verbs in a sentence, "DeepSLN" model exploit the logic rule based on the number of nouns, and finally "DeepSLVN" use both of the logic rules.

The results confirm our intuition since "DeepSLVN" outperformed the two previous models based on just one logic rule. We show the experimental results in the first three rows of Table 5. Moreover, we evaluate our models against the previous best models, and we summarize the experiments in Table 5 where we were able to outperform the two best models. From the previous results, we observe an essential remark which is the

Model	Accuracy
DeepSLV	97.59
DeepSLN	97.61
DeepSLVN	**97.69**
[43]	97.50
[9]	97.55

Table 5: POS tagging accuracy of our system on test data from WSJ proportion of PTB together with a comparison to the best previous systems.

ability of our two models ($DeepSLV$ and $DeepSLN$) to outperform the baseline [9] by adding one logic rule, which confirms the importance of combining logic rules with deep neural networks for other NLP tasks such POS tagging.

5 Conclusion and Future Work

In this paper, we showed that the uninterpretability problem faced by deep neural networks could be reduced by adding logic rules in the form of SL and PSL rules. We demonstrate that adding more logic rules is useful because it helped the overall framework to improve the performance. The experiments confirm these results on the sentiment classification task. Moreover, we apply this framework to a new NLP task namely POS tagging where we were able to improve the performance of our system.

In the future work, we will explore the negation relation, which we believe it will be very useful for all sentences containing a negation word such as "cannot", "never", "hardly", "nothing", "neither", etc., in this case, the segment's sentiment is flipped. Furthermore, we will also explore the non-contrast relations and combine them in one framework. Moreover, we will apply our framework to other languages (French and German) for more generalization.

References

1. Hu, Z., Ma, X., Liu, Z., Hovy, E., Xing, E.: Harnessing deep neural networks with logic rules. arXiv preprint arXiv:1603.06318 (2016)
2. Dahl, G.E., Yu, D., Deng, L., Acero, A.: Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Transactions on Audio, Speech, and Language Processing **20** (2012) 30–42
3. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al.: Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine **29** (2012) 82–97
4. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR. Volume 1. (2017) 3
5. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al.: Mastering the game of go without human knowledge. Nature **550** (2017) 354

6. Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259 (2014)
7. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. (2014) 3104–3112
8. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. (2013) 3111–3119
9. Ma, X., Hovy, E.: End-to-end sequence labeling via bi-directional lstm-cnns-crf. arXiv preprint arXiv:1603.01354 (2016)
10. Gridach, M.: Character-level neural network for biomedical named entity recognition. Journal of biomedical informatics **70** (2017) 85–91
11. Gridach, M.: Character-aware neural networks for arabic named entity recognition for social media. In: Proceedings of the 6th workshop on South and Southeast Asian natural language processing (WSSANLP2016). (2016) 23–32
12. Gridach, M., Haddad, H.: Arabic named entity recognition: A bidirectional gru-crf approach. In: International Conference on Computational Linguistics and Intelligent Text Processing, Springer (2017) 264–275
13. Gridach, M., Haddad, H., Mulki, H.: Churn identification in microblogs using convolutional neural networks with structured logical knowledge. In: Proceedings of the 3rd Workshop on Noisy User-generated Text. (2017) 21–30
14. Amiri, H., Daume III, H.: Short text representation for detecting churn in microblogs. In: Thirtieth AAAI Conference on Artificial Intelligence. (2016)
15. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
16. Gridach, M., Haddad, H., Mulki, H.: Empirical evaluation of word representations on arabic sentiment analysis. In: International Conference on Arabic Language Processing, Springer (2017) 147–158
17. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Cognitive modeling **5** (1988) 1
18. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
19. Minsky, M.: Learning meaning. Artificial Intelligence Laboratory, Massachusetts Institute of Technology (1983)
20. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Advances in Neural Information Processing Systems. (2017) 3859–3869
21. Bach, S.H., Broecheler, M., Huang, B., Getoor, L.: Hinge-loss markov random fields and probabilistic soft logic. arXiv preprint arXiv:1505.04406 (2015)
22. Donadello, I., Serafini, L., Garcez, A.d.: Logic tensor networks for semantic image interpretation. arXiv preprint arXiv:1705.08968 (2017)
23. Kouki, P., Fakhraei, S., Foulds, J., Eirinaki, M., Getoor, L.: Hyper: A flexible and extensible probabilistic framework for hybrid recommender systems. In: Proceedings of the 9th ACM Conference on Recommender Systems, ACM (2015) 99–106
24. Sridhar, D., Pujara, J., Getoor, L.: Using noisy extractions to discover causal knowledge. arXiv preprint arXiv:1711.05900 (2017)
25. Kimmig, A., Bach, S., Broecheler, M., Huang, B., Getoor, L.: A short introduction to probabilistic soft logic. In: Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications. (2012) 1–4
26. Ganchev, K., Gillenwater, J., Taskar, B., et al.: Posterior regularization for structured latent variable models. Journal of Machine Learning Research **11** (2010) 2001–2049
27. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)

28. Nakashole, N., Flauger, R.: Knowledge distillation for bilingual dictionary induction. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. (2017) 2497–2506
29. Zhu, X., Gong, S., et al.: Knowledge distillation by on-the-fly native ensemble. In: Advances in Neural Information Processing Systems. (2018) 7527–7537
30. Chen, G., Choi, W., Yu, X., Han, T., Chandraker, M.: Learning efficient object detection models with knowledge distillation. In: Advances in Neural Information Processing Systems. (2017) 742–751
31. Tang, J., Wang, K.: Ranking distillation: Learning compact ranking models with high performance for recommender system. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM (2018) 2289–2298
32. Ma, X., Zhao, H.: Probabilistic models for high-order projective dependency parsing. arXiv preprint arXiv:1502.04174 (2015)
33. Ma, X., Hovy, E.: Efficient inner-to-outer greedy algorithm for higher-order labeled dependency parsing. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. (2015) 1322–1328
34. Ma, X., Liu, Z., Hovy, E.: Unsupervised ranking model for entity coreference resolution. arXiv preprint arXiv:1603.04553 (2016)
35. Wang, Y., Wu, S., Li, D., Mehrabi, S., Liu, H.: A part-of-speech term weighting scheme for biomedical information retrieval. Journal of biomedical informatics **63** (2016) 379–389
36. Schlünz, G.I., Dlamini, N., Kruger, R.P.: Part-of-speech tagging and chunking in text-to-speech synthesis for south african languages. In: INTERSPEECH. (2016) 3554–3558
37. Fakhraei, S., Huang, B., Raschid, L., Getoor, L.: Network-based drug-target interaction prediction with probabilistic soft logic. IEEE/ACM Transactions on Computational Biology and Bioinformatics **11** (2014) 775–787
38. Socher, R., Perelygin, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C., et al.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP). Volume 1631. (2013) 1642
39. Pang, B., Lee, L.: Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd annual meeting on association for computational linguistics, Association for Computational Linguistics (2005) 115–124
40. Xiao, L., Zhang, H., Chen, W., Wang, Y., Jin, Y.: Transformable convolutional neural network for text classification. In: IJCAI. (2018) 4496–4502
41. Zhang, Y., Roller, S., Wallace, B.: Mgnc-cnn: A simple approach to exploiting multiple word embeddings for sentence classification. arXiv preprint arXiv:1603.00968 (2016)
42. Ouyang, X., Gu, K., Zhou, P.: Spatial pyramid pooling mechanism in 3d convolutional network for sentence-level classification. IEEE/ACM Transactions on Audio, Speech, and Language Processing **26** (2018) 2167–2179
43. Søgaard, A.: Semisupervised condensed nearest neighbor for part-of-speech tagging. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2, Association for Computational Linguistics (2011) 48–52