An Exploration of Reinforced Learning Based Representation on Hate Speech Detection

CSI 5386 Wei Li Libo Long

Abstract

Using reinforcement learning based structure is a novel idea in natural language processing. Tianyang Zhang and Minlie Huang proposed Information Distilled LSTM (ID-LSTM) and Hierarchical Structured LSTM (HS-LSTM) which learn policies to simplify information and learn structured representation to sentences. In this paper, we trained the two models in Offensive Language Identification task and compare their performance with other baselines. We also add BiLSTM, attention and BERT components to the architecture in ID-LSTM and HS-LSTM. The result shows that although the original ID-LSTM and HS-LSTM only slightly extend the baselines, their modified version does significantly better. The result indicates that reinforcement learning structures have more potential in natural language processing. The code to this project is available in REPO.

1. Introduction

A foundation problem in many natural language process tasks is learning good representation to texts (i.e. word embedding, sentence embedding, document embeddings). In recent representation learning researches, a series of neural network methods have been proposed. The mainstream architecture can be classified roughly into 4 classes: 1) convolution network that learns representation through convolution and pooling layers, 2) recurrent network that considers the word orders and 3) structured RNN like tree-LSTM(Tai et al., 2015)(Tai et al., 2015) which also considers the structure of the context from pre-defined parsing. More recently, attention and transformer-based models achieved the state of arts performance.

Tianyang Zhang and Minlie Huang proposed a novel method(Zhang et al., 2018a) that uses reinforcement learning method to explore and identify task-related structures automatically. Based on the Reinforcement learning structure, they presented Information Distilled LSTM (ID-LSTM), which learns a strategy to delete unrelated words in texts, and Hierarchical Structured LSTM (HS-LSTM), which learns a strategy to group words into sub-phrases. The two models both show comparable and better performance to baseline CNN, LSTM, Self-Attentive and tree-LSTM models in classification tasks. The structure is also easy to be modified by introducing more task-related policy space and using other representation networks.

Hate speech detection is an application of text classification and is deployed on many social media platforms. The objective of hate speech detection is identifying abusive languages that target specific individuals or groups. Many datasets have been proposed to solve the problem and many of which are based on Twitter corpora.

In this project, we consider the downstream classification task in Offensive Language Identification Dataset (OLID)(Zampieri et al., 2019a), implement and extend the ID-LSTM and HS-LSTM models, and compare them with mainstream baseline models. The result shows that after adding more recent natural language processing structures, the two reinforcement learning based models gain significantly better performance compared to the baselines.

2. Related work

In Natural Language Processing, the classification task is to assign tags to texts, given information extracted from the texts. Text classification is one of the fundamental tasks in Natural Language Processing and is widely used in many applications such as sentimental analysis, topic classification, spam detection, author classification, etc. Chronologically, 3 groups of methods have been used in text classification: rule-based approaches, machine learning approaches, and deep learning approaches. Rule-based approaches use a set of rules summarized from linguistic experiences and researches. These rules are handcrafted for each specific task and dataset and required a lot of domain knowledge, which render the approach less economical comparing to latter approaches. A more recent approach, Machine learning systems trying to extract features from the unstructured text data and apply machine learning to learn a distribute that map the feature representations to tags. One of the most commonly used feature extraction methods is bag-of-words, which represent the text by counting the frequency of words

appearing in a corpus. In recent years, deep learning architectures achieved state of art results in text classification. In deep learning, representation of text can be learned automatically and effectively from corpora, and the result can be generalized well into different tasks.

2.1. Convolutional Neural Networks (CNN)

CNN(Kim, 2014)(Kalchbrenner et al., 2014): Although originally built for image processing, CNN also shows effectiveness for text classification. Text CNN shares the same architecture as image processing. The input images or texts are convolved with a set of kernels of size $d \times d$. A convolution layer is also called a feature map and they can be stacked up to form a more complex filter. CNN structure also uses pooling techniques. Pooling is introduced to reduce the complexity and dimension introduced by convolution. The most common pooling methods are max pooling and mean pooling where the maximum element or the average of the pooling window is kept in the next layer.

In Natural Language Processing, CNN is usually implemented in the following way: a sentence is embedding into a matrix, which is passed into convolution layers of different kernels size. The outputs of convolution layers are then passed into pooling layers, and the outputs are concatenated to a single vector, which is the vectorized representation of the sentence.

2.2. Recurrent Neural Network (RNN)

RNN (Hochreiter & Schmidhuber, 1997)(Chung et al., 2014): Recurrent Neural Network is the most widely used neural network architecture for sequential data type like texts. RNN treats inputs as a sequence, and assign weights to the previous state in a sequence, which make it able the memorize the information form the previous state, and thus become an ideal architecture for sequential data, strings and sequential data classification. A vanilla RNN uses a simple affine function to calculate the state parameters, but the structure is vulnerable to the vanishing gradient and the exploding gradient problem, especially when the input sentence is long. The Recurrent Neural Network is more often implemented as LSTM or GRU, which introduced the gate mechanism to cope with the vanishing gradient and the exploding gradient problem.

2.3. Long Short-Term Memory (LSTM) and BiLSTM

Long Short-Term Memory(LSTM) (Hochreiter & Schmidhuber, 1997): is a special RNN with additional gated structure. LSTM is good at addressing vanishing gradient problem since it allows long term information to preserve through a long sentence. Unlike GRU, LSTM uses more complex gates to carefully modify the amount of information that goes into each state.

In LSTM, information only goes in one direction (from the beginning of a sentence to the end of the sentence). However, in a sentence, important information to an early state may also occur in the latter part of the sentence. BiLSTM uses two LSTMs that flow in different directions and combines the hidden state to produce an encoding that considers information in both directions.

Finally, to make the model more complex, the LSTM layers can be stacked up to create multi-layer LSTM models

2.4. Hierarchical Attention Networks

Hierarchical Attention Networks(Yang et al., 2016): this method has a hierarchical structure that mirrors the hierarchical structure of documents; The member of the hierarchy can be words, sentences and documents. Moreover, each hierarchy has its attention. A hierarchy structure of document-level classification can begin with a word encoder with word-level attention, which is feed into a sentence encoder with sentence-level attention, and the output of the pooling of the sentence encoding is the document-level encoding for the document. A Hierarchical Attention Networks of a document classification problem is shown in Figure 1.

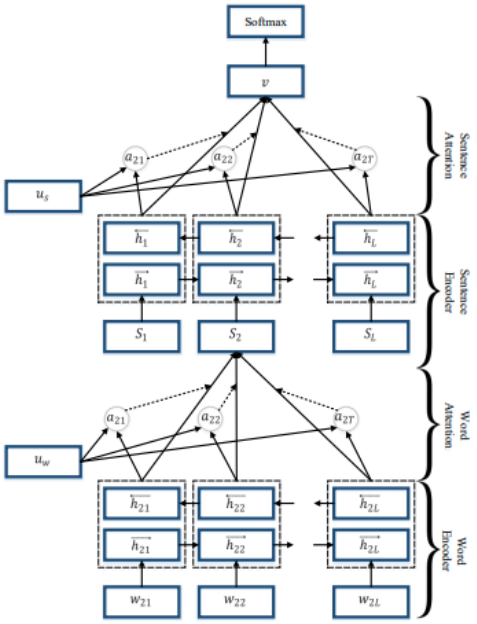

Figure 1. Hierarchical Attention Networks for Document Classification

2.5. ID-LSTM and HS-LSTM

Information Distilled LSTM (ID-LSTM) and Hierarchical Structured LSTM (HS-LSTM). (Zhang et al., 2018a) are two novel sentence representation models that, unlike the architecture mentioned above, use reinforcement learning-based technique to explore and identify task-related structures automatically, and learning more task-specific sentence representation.

The two models both consist of three components: a Policy Network(PNet), a Structure Representation Model, and a classification Network(CNet).

Firstly, PNet learns a policy and samples an action in each state, which will modify the structure of the input sentence. The modification will be reflected by the sentence encoding in the structured representation model. The Structured Representation Model takes the action and produce a vector representation of the text. The vector is then fed into CNet, a multilayer perceptron to produce the classification result. Finally, the reward to the PNet is calculated from the classification distribution of the CNet.

Based on the choice of different policy spaces, the structure can derive different models that learn different text structures. In the proposed Information Distilled LSTM, the action space is Retain, Delete, where a word can be deleted from or retained in the final sentence representation. In Hierarchical Structured LSTM, the action space is Inside, End, which groups words into sub-phrases: if a word does not finish a phrase, the PNet return 'Inside', if a word finishes a phrase, the PNet return 'End'.

The Information Distilled LSTM and Hierarchical Structured LSTM is proposed in 2018. In the original paper, The word vectors are initialized using 300- dimensional Glove vectors and single directional LSTM is choice as the Structured Representation Model. In this paper, we implement the ID-LSTM and HS-LSTM with some modifications that incorporate recent widely used structures like BERT and attention, and compare the result of the modified models with the original model and other baselines.

3. Methodology

We firstly introduce the implementation of the original Information Distilled LSTM and Hierarchical Structured LSTM (Figure 1) and then introduce each of the modifications we add to the original models.

3.1. Information Distilled LSTM (ID-LSTM)

The objective of Information Distilled LSTM is to build a sentence representation by removing irrelevant words in a sentence and preserve important words. Given a sentence embedded as a list of vectors $X = \mathbf{x_1 x_2} \cdots \mathbf{x_L}$, the LSTM in the Structure Representation Model take the t-th word $\mathbf{x_t}$ and calculate the hidden state $\mathbf{h_t}$ and cell state $\mathbf{c_t}$. The concatenation of the last hidden state, cell state and current word vector is the state for the agent in the policy net:

$$\mathbf{s_t} = \mathbf{c_{t-1}} \oplus \mathbf{h_{t-1}} \oplus \mathbf{x_t} \tag{1}$$

where $\oplus$ indicates vector concatenation and $\mathbf{x_t}$ is the current word input.

PNet is a double-layer LSTM model. It takes a series of state $\mathbf{s_t}$ as input and outputs a series of action $A = \mathbf{a_1 a_2} \cdots \mathbf{a_L}$. For ID-LSTM, the action space is {Retain, Delete}, where Retain indicates that the word is retained in a sentence, and Delete means that the word is deleted and it has no contribution to the final encoding. Formally, if PNet produce a $\mathbf{a_t}$ = *Retain* for the t-th word, then the hidden state $\mathbf{h_t}$ and cell state $\mathbf{c_t}$ of the words will be calculate from the current word and the last $\mathbf{h_{t-1}}$ and $\mathbf{c_{t-1}}$. If PNet produce a $\mathbf{a_t}$ = *Delete*, then the current hidden state $\mathbf{h_t}$ and cell state $\mathbf{c_t}$ will be a direct copy of the last hidden state $\mathbf{h_{t-1}}$ and cell state $\mathbf{c_{t-1}}$

$$\mathbf{c_t}, \mathbf{h_t} = \begin{cases} \mathbf{c_{t-1}}, \mathbf{h_{t-1}}, & a_t = \text{Delete} \\ \Phi\left(\mathbf{c_{t-1}}, \mathbf{h_{t-1}}, \mathbf{x_t}\right), & a_t = \text{Retain} \end{cases} \tag{2}$$

In other word, $\mathbf{x_t}$ has not contribution to the information that is passed down the network.

CNet is a four-layer fully connected network, which takes vectorized sentence representation produced by Structure Representation LSTM as input and add Softmax to produce the predicted distribution.

$$P(y|X) = \text{softmax}\left(\mathbf{W_s h_L} + \mathbf{b_s}\right) \tag{3}$$

where $\mathbf{W_s} \in \mathbb{R}^{d \times K}, \mathbf{b_s} \in \mathbb{R}^K$ are parameters of CNet, d is the dimension of hidden state, $y \in \{c_1, c_2, \cdots, c_K\}$ is the class label and K is the number of categories.

The reward to the PNet is calculated from the logarithm of the probability of correct CNet classification plus a regularizing term controlling how many words are deleted. i.e.

$$R_L = \log P\left(c_g|X\right) + \gamma L'/L \tag{4}$$

where $\mathcal{C}_g$ is the gold label of the input X and L' denotes the number of deleted words. γ is a hyper-parameter to balance the two terms. The reward to the PNet is a delayed reward, which requires the three models to be trained jointly.

3.2. Hierarchical Structured LSTM (HS-LSTM)

Hierarchical models are widely used in document-level classification. The Hierarchical Structured LSTM is inspired

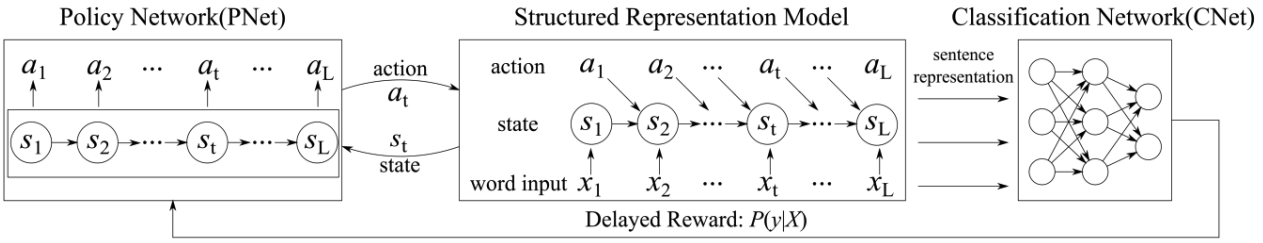

Figure 2. PNet outputs an action at each state. The structured representation model offers state representation to PNet and outputs the final sentence representation to the CNet when all actions are sampled. CNet performs text classification and provides a reward to PNet.

by the Hierarchical Attention Networks (Yang et al., 2016) structure. It is built to discover hierarchical sub-structures of a sentence and to construct structured sentence representation.

Like Information Distilled LSTM, PNet in Hierarchical Structured LSTM is also a double-layer LSTM model. taking a series of state $\mathbf{s_t}$ as input and outputs a series of action $A = \mathbf{a_1 a_2} \cdots \mathbf{a_L}$. For Hierarchical Structured LSTM, the action space is {Inside, End}, where Inside indicates that a word is inside of a phrase and End means the word is at the end of a phrase. An example sequence of action is {Inside, Inside, Inside, End, Inside, Inside, End}, meaning that the first four words form a sub-phrase, and the last three words form another sub-phrase.

In HS-LSTM, the LSTM of the Structure Representation Model is a two-level structure: a word-level LSTM which use PNet to form a sub-phrase, and a phrase-level LSTM which connects those sub-phrases to form a sentence representation.

Formally, if PNet produce an $\mathbf{a_t} = \textit{End}$ for the t-th word, i.e. the word $\mathbf{x_{t+1}}$ begins a new sub-phrase, then the word level LSTM will start with a zero-initialized state. If PNet produce an $\mathbf{a_t} = \textit{Inside}$ for the t-th word, the sub-phrase does not end, thus the word-level LSTM continue from the previous word in the phrase. The process is defined as follow:

$$\mathbf{c^w_{t+1}}, \mathbf{h^w_{t+1}} = \begin{cases} \Phi^w\left(\mathbf{0}, \mathbf{0}, \mathbf{x_t}\right), & \mathbf{a_{t-1}} = \textit{End} \\ \Phi^w\left(\mathbf{c^w_t}, \mathbf{h^w_t}, \mathbf{x_t}\right), & \mathbf{a_{t-1}} = \textit{Inside} \end{cases} \tag{5}$$

where Φ^w denotes the transition functions of the word-level LSTM, $\mathbf{c_t}$ is the memory cell, and $\mathbf{h_t}$ is the hidden state at position t.

The input of the phrase-level LSTM depend on action $\mathbf{a_t}$. If action $\mathbf{a_t}$ is End, which means the current position t completes a phrase, the hidden state from the word-level LSTM in position t (i.e. $\mathbf{h^w_t}$) will be fed into phrase-level LSTM. Otherwise, when action $\mathbf{a_t}$ is Inside, the phrase-level LSTM will be frozen until it meet another End, and all the variables are copied from the previous position. Formally,

$$\mathbf{c^p_t}, \mathbf{h^p_t} = \begin{cases} \Phi^p\left(\mathbf{c^p_{t-1}}, \mathbf{h^p_{t-1}}, \mathbf{h^w_t}\right), & a_t = \textit{End} \\ \mathbf{c^p_{t-1}}, \mathbf{h^p_{t-1}}, & a_t = \textit{Inside} \end{cases} \tag{6}$$

where Φ^p denotes the transitions function of the phrase-level LSTM. Note that the input to the phrase-level LSTM is $\mathbf{h^w_t}$, the hidden state of the word-level LSTM. The Structure Representation Model in HS-LSTM can be illustrate by Figure 3.

Given a sentence embedded as a list of vectors $X = \mathbf{x_1 x_2} \cdots \mathbf{x_L}$, the word-level LSTM in the Structure Representation Model takes the t-th word $\mathbf{x_t}$ and calculate the hidden state $\mathbf{h^W_t}$ and cell state $\mathbf{c^W_t}$. The two states are concatenated with cell state and hidden state from the previous phrase network, which forms the state for the PNet.

$$\mathbf{s_t} = \mathbf{c^p_{t-1}} \oplus \mathbf{h^p_{t-1}} \oplus \mathbf{c^w_t} \oplus \mathbf{h^w_t}$$

CNet is a four-layer fully connected network, which takes the last hidden state of the phrase-level LSTM (h^P_L) as input and add Softmax to produce the predicted distribution.

$$P(y|X) = \text{softmax}\left(\mathbf{W_s h_L} + \mathbf{b_s}\right) \tag{7}$$

Like the ID-LSTM, the reward is also based on how prediction matches gold labels. Unlike ID-LSTM's reward which encourages the model to remove as many words as possible, reward of HS-LSTM respects that a good phrase structure should contain neither too many nor too few phrases, which corresponding to a regularizing term using unimodal function. The reward function is calculated as follows.

$$R_L = \log P\left(c_g|X\right) - \gamma\left(L'/L + 0.1L/L'\right)$$

where L denotes the number of phrases (the number of action End). γ is a hyper-parameter. The second term encourages the number of phrases to be $0.316L$

3.3. Extension

The original paper used GloVe word embedding and LSTM in the architecture. Based on other popular sentence representation architecture and word-level representation, we

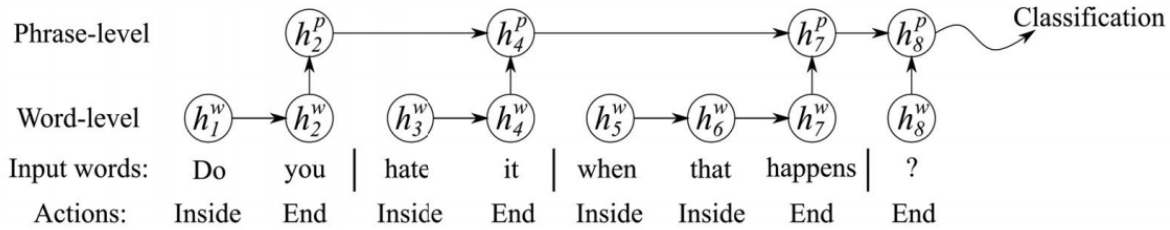

Figure 3. Structure Representation Model in HS-LSTM

extend the original model by incorporating BERT, attention layers and BiLSTM in the Information Distilled(ID) and Hierarchical Structured(HS) structure. Concretely, we implemented the following 12 models:

- **ID-LSTM and HS-LSTM**: the original models as described above.
- **ID-LSTM and HS-LSTM + attention**: we add an attention layer to the Policy Network and Structured Representation Model in both Information Distilled and Hierarchical Structured architecture. (Figure 4)
- **ID-BiLSTM and HS-BiLSTM**: We change the Structured Representation Model to double-layer BiLSTM in both Information Distilled and Hierarchical Structured architecture. (Figure 5)
- **ID-BiLSTM and HS-BiLSTM + attention**: We change the layer of Structured Representation Model to double-layer BiLSTM and add attention layers to Policy Network and Structured Representation Model in both Information Distilled and Hierarchical Structured architecture. (Figure 6)
- **Pre-trained BERT + ID-LSTM and HS-LSTM**: We add Pre-trained BERT (Devlin et al., 2019) to embed the dataset before Information Distilled and Hierarchical Structured architecture. (Figure 7)
- **Pre-trained BERT + ID-BiLSTM and HS-BiLSTM**: We add Pre-trained BERT (Devlin et al., 2019) to embed the dataset before Information Distilled and Hierarchical Structured architecture and change Structured Representation Model to double-layer BiLSTM. (Figure 8)

4. Experiment Setup

We test the representation models on the downstream classification task. We chose Hate Speech Detection as the specific task. Hate Speech Detection is a classification task that classifies whether texts contain hate speech or abusive language. More advanced Hate Speech Detection task includes identifying the type of hate speech, whether the hate speech has a target and identifying the target. Twitter is the most widely used data source in abusive language research.

4.1. Dataset

We use Offensive Language Identification Dataset (OLID) v1.0 (Zampieri et al., 2019b) as the dataset. The OLID contains 14,100 annotate tweets which has been used as the official dataset for OffensEval(Zampieri et al., 2019a).

In the OLID dataset, there are 3 levels of sub-tasks. The first level (level-a) is Offensive language identification - a binary classification of whether a tweet contains offensive language. The second and the third level (level-b and level-c) are 'categorization of offense types' and 'offense target identification' respectively. Since the level-b and level-c require the tweet to be offensive in the first place, the two sub-tasks contain much lesser data points. In the experiment, we only consider the level-a sub-task.

The classes distribution of the level-a is balanced, about half of the tweets are Offensive and the reminder are Not-Offensive. We use 13,240 tweets as the training set and the remainder as the test set.

4.2. Baselines

Beside the 12 Reinforced Learning models, we also trained 2 baselines models as reference, they are:

- **BiLSTM**: A bi-directional LSTM, commonly used in text classification (Hochreiter & Schmidhuber, 1997)
- **CNN**: Convolutional Neural Network (Kim, 2014)

4.3. Preprocessing and training

Tweets in OLID contain lots of emojis, typos, and texts other than English. Therefore, we use TweetTokenizer from NLTK (Bird et al., 2009) to tokenize all tweets. We modified the tokenizer so that all the '@user' were replaced by token '<user>', all the web URLs were replaced by token '<url>', and all twitter hashtags '#topic' were replaced by

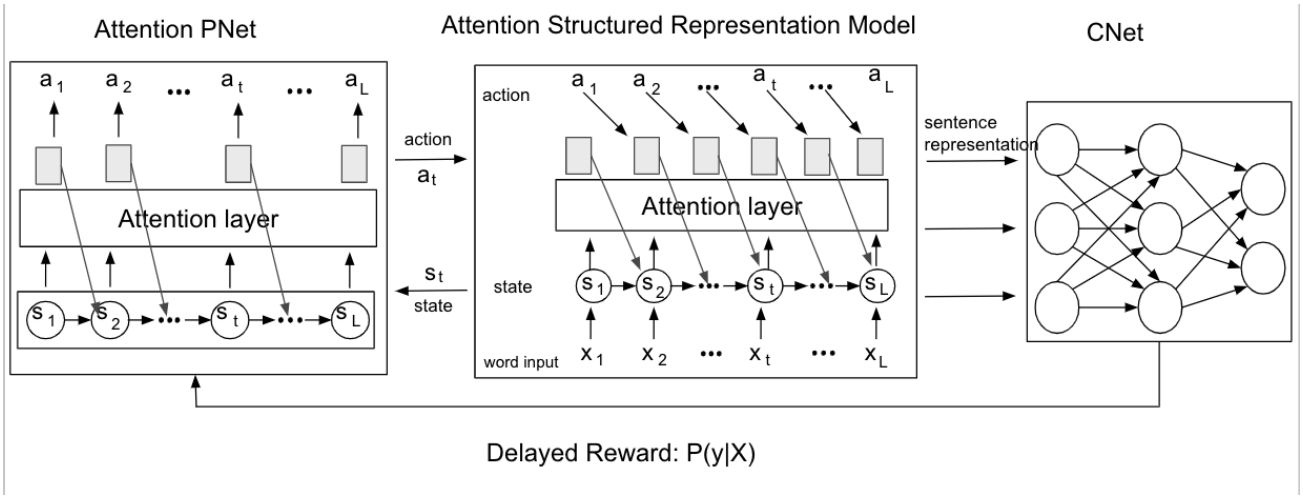

Figure 4. ID-LSTM and HS-LSTM + attention Structure

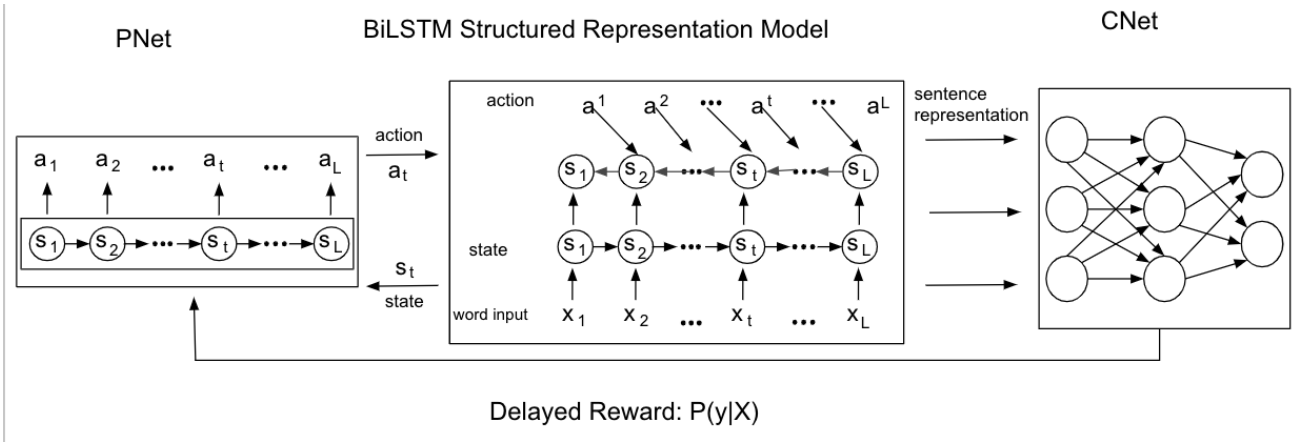

Figure 5. ID-BiLSTM and HS-BiLSTM Structure

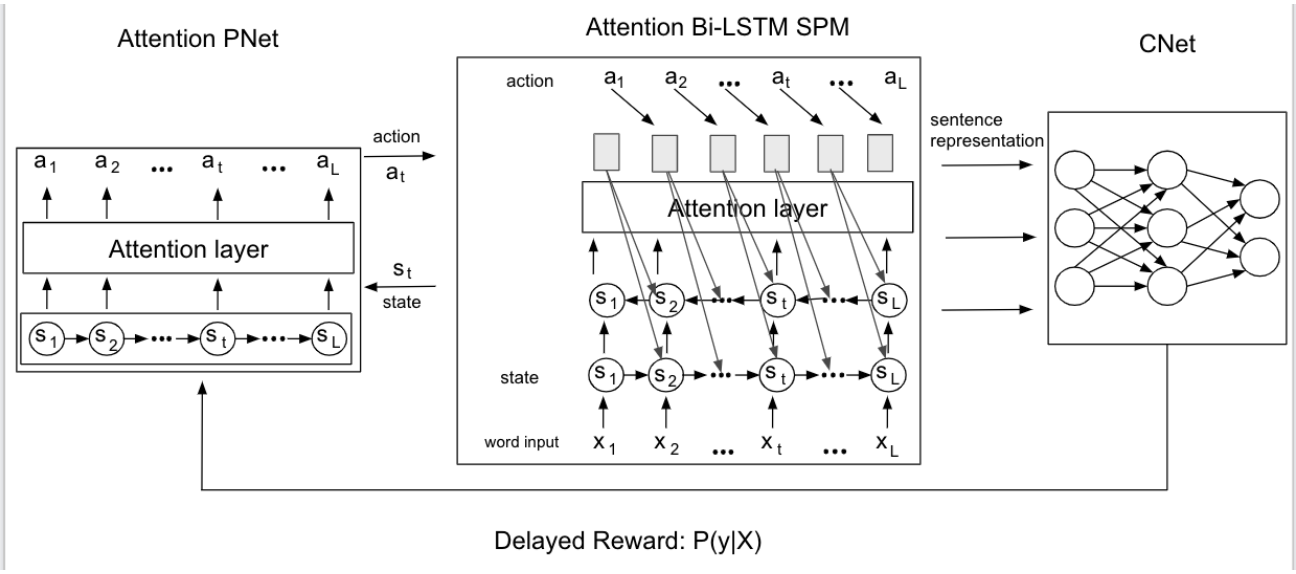

Figure 6. ID-BiLSTM and HS-BiLSTM + attention Structure

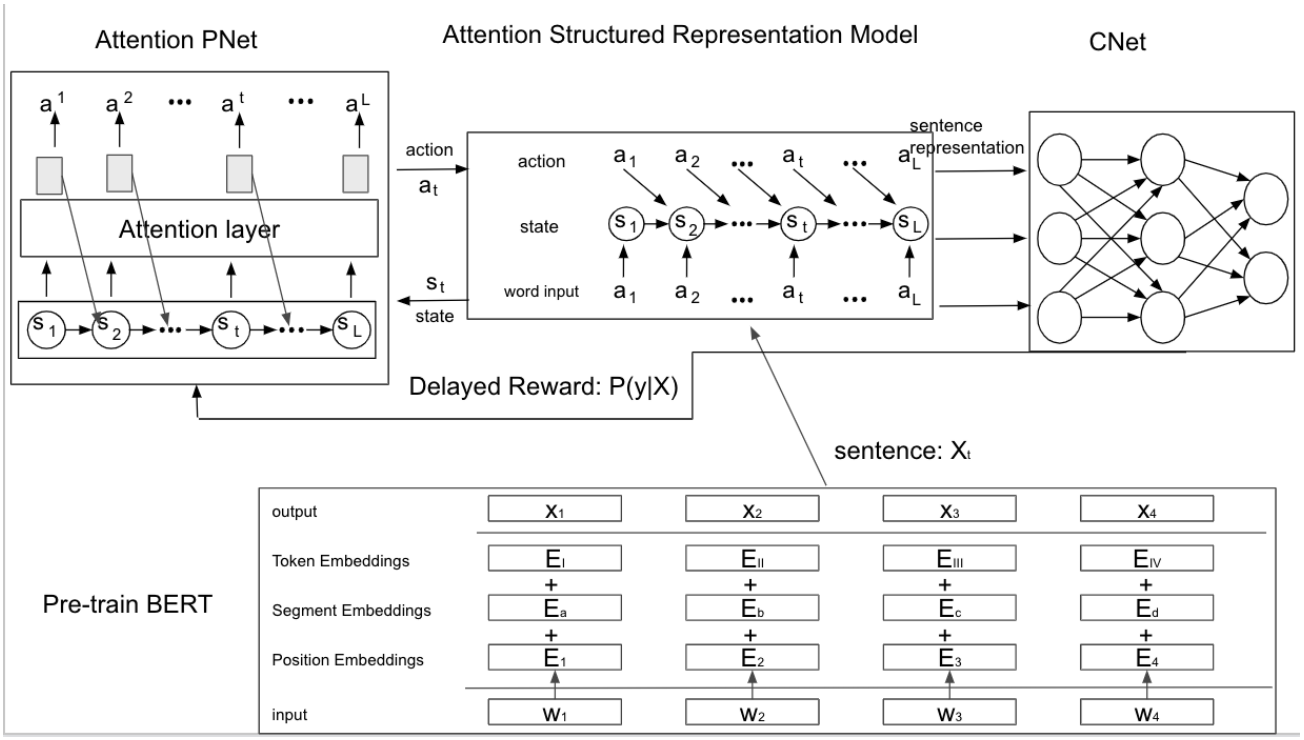

Figure 7. Pre-trained BERT + ID-LSTM and HS-LSTM Structure

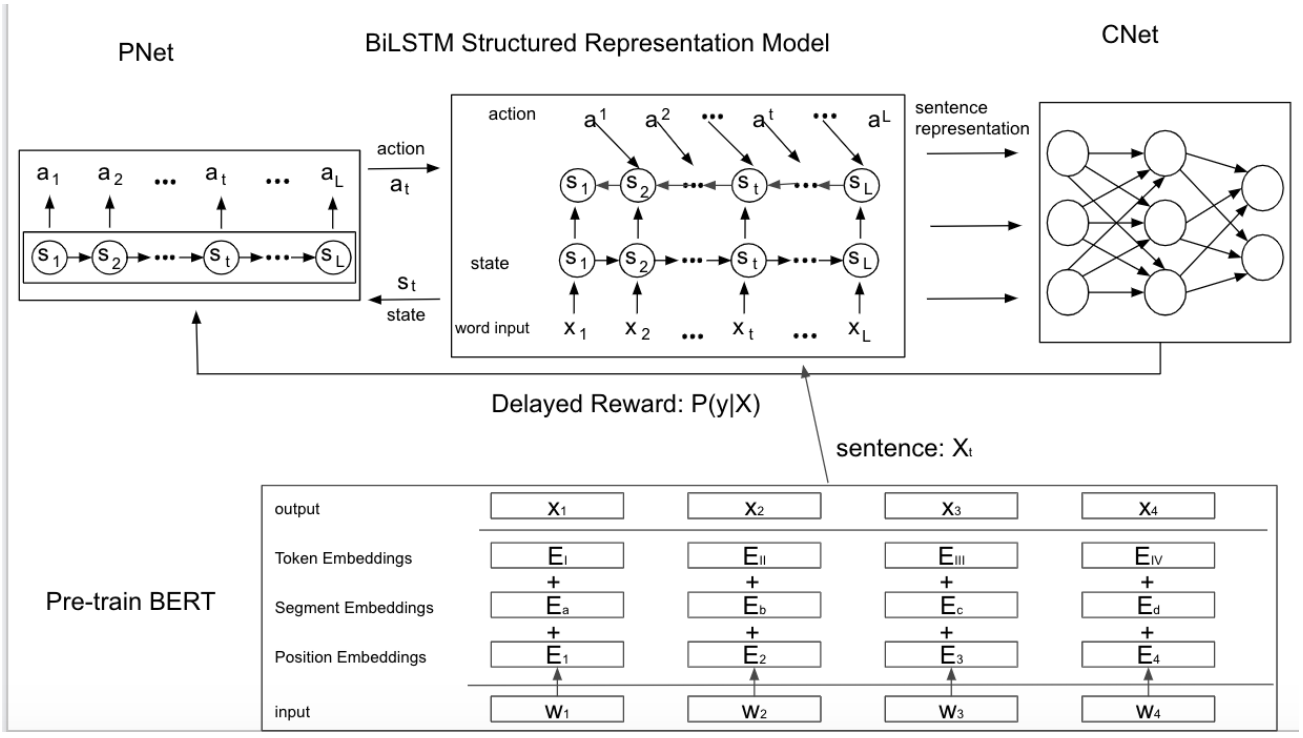

Figure 8. Pre-trained BERT + ID-BiLSTM and HS-BiLSTM Structure

token '<hashtag>'. We ignore all foreign languages.

For experiments that do not involve using the BERT-pretrained model, we use 200-dimensional GloVe word embedding(Pennington et al., 2014) that pretrained on 2B tweets to embed words, and use 300-dimensional Emoji2Vec(Eisner et al., 2016) pretrained embedding to embed emojis. To keep the dimension consistent, we use Principal Component Analysis(PCA) to embed emoji vectors in a 200-dimensional vector space.

For training reinforcement learning based models, firstly, we train Structure Representation Model and CNet using the original sentence without any modification - the process is the same as the LSTM classification. Then, we fix all parameters in C-Net and train PNet, which learns to adopt actions to the input sentence. Finally, we train PNet and CNet together.

In all experiments, we use Adam optimizer with a 0.0005 learning rate. Baseline models are trained 100 epochs, while all Reinforce Learning based models, given their computation complexity, are trained 50 epochs. We use learning rate decay that decreases learning rate in the last 10 epochs. We adopt dropout before the classification network with a probability of 0.5. Also, the γ in the reward function is set as 0.15 for Information Distilled models and 0.1 for Hierarchical Structured models.

4.4. Evaluation

Although the 2 class in OLID dataset is balanced, one observation is made in Hate Speech Detection that there exist some bias to certain groups of user, and thus models tend to generate false-positive results (Davidson et al., 2019). For example, it is innocuous in the context of the homosexual community to assert "I am a gay man", but when the statement is evaluated in a big data set together with comments from other groups, it gets high toxicity score.

Based on the problem, we consider include more evaluation metrics that take false-positive results into account. The evaluation metrics we use are:

- **Accuracy**: the performance of each model in a general sense
- **Precision**: taking the false positive problem in hate speech detection task into consideration, we calculate precision for both class
- **F1 score**: a balanced evaluation that takes both bias and performance into account, we calculate precision for both class
- **Recall**: a supplementary metric, we calculate Recall for both classes

To reduce variance in the final result, in each of the training steps of the last 5 epochs, we sample all data points in the test set and evaluate the results. All the evaluation metrics are also calculated and stored. The final result is the mean value of all the results.

5. Result Analysis

The experiment result is given in Table 1. The overall performance of the ID-LSTM and HS-LSTM without modification is identical or slightly better than the baseline models CNN and BiLSTM, an observation that aligns with the result from the original paper(Zhang et al., 2018b). However, the modified version of ID-LSTM and HS-LSTM get even better performance from the original model. Comparing to BiLSTM, modified ID-BiLSTM+attention and HS-BiLSTM+attention increase by about 1% in accuracy and about 30% in the recall rate of offensive tweets.

BERT pre-trained models get the highest classification accuracy and are generally better than the same structure that uses GloVe and Emoji2Vec embedding. Of all implemented models, BERT+ID-BiLSTM gets the best result.

It also observed that compares to baseline models, ID models and HS models' precision to Offensive tweets decreased 5% to 10%, but the recall of Offensive tweets is increased by 30%. The result indicates that the ID and HS structure sacrificed some Offensive language precision rate for higher Offensive language Recall rate. Also, it is observed that the false-positive problem is not very severe since the precision of Offensive class is relatively high.

Also, while the Non-offensive class F score is comparable to the baseline, the Offensive class F score of the reinforced learning models are significantly better. Overall, Information Distilled models and Hierarchical Structured models are better than the baseline.

Comparing Information Distilled models and Hierarchical Structured models, it is observed that the two types of models get similar results. The Hierarchical Structured models are slightly better than Information Distilled models, especially in the recall rate of Offensive tweets. A possible reason is that tweets in the OLID dataset are prone to be short and condense in meaning. The Information Distilled models' deleting mechanism may not be helpful in short sentences. Hierarchical Structured models' good performance indicates that discovering hierarchical structures for short sentences may help to improve the prediction. An Example tweet after Information Distilled and Hierarchical Structured models is given in Figure 9.

Model	Accuracy	Precision NOT	Precision OFF	Recall NOT	Recall OFF	F score NOT	F score OFF
CNN	81.09	41.19	70.24	90.83	55.92	87.37	62.14
BiLSTM	**82.80**	**84.17**	**77.40**	**93.80**	**54.42**	**88.70**	**63.68**
ID-LSTM	82.19	91.47	64.53	83.06	79.92	87.07	71.40
ID-LSTM+attention	83.00	92.47	65.45	83.23	82.43	87.61	72.96
ID-BiLSTM	82.77	**93.70**	64.26	81.61	**85.77**	87.24	73.48
ID-BiLSTM+attention	83.35	92.21	66.33	84.03	81.59	87.93	73.17
BERT+ID-LSTM	85.22	92.57	**70.00**	**86.45**	82.01	89.41	75.53
BERT+ID-BiLSTM	**85.56**	93.66	69.76	85.81	84.94	**89.56**	**76.60**
HS-LSTM	83.35	93.28	65.58	82.90	84.52	87.79	73.86
HS-LSTM+attention	83.93	93.66	66.45	83.39	85.36	88.23	74.73
HS-BiLSTM	83.47	91.07	67.51	85.48	78.24	88.19	72.48
HS-BiLSTM+attention	84.05	93.51	66.78	83.71	84.94	88.34	74.77
BERT+HS-LSTM	87.08	94.26	72.54	87.42	**86.19**	90.71	78.78
BERT+HS-BiLSTM	**87.31**	**94.28**	**73.05**	**87.74**	**86.19**	**90.89**	**79.08**

Table 1. Results of the experiment, the first part is baseline models; the second part is ID-LSTM and extensions; the third part is HS-LSTM and extensions. NOT stands for not offensive, OFF stands for offensive

6. Conclusion and Future Work

In this paper, we successfully implemented reinforcement learning based Information Distilled LSTM, Hierarchical Structured LSTM and a variety of their modification. The test on the Offensive Language Identification Dataset (OLID) shows that while the original ID-LSTM and HS-LSTM achieve comparable or slightly better performance than the baselines, adding more recent structures to the model can help it get a more significant boost in performance. The result confirms that the two structures have good potential to be incorporated with more complex elements and to achieve a better result.

Based on this study, we can experiment with other action spaces and structures other than LSTMs. Also, given the close relationship of hierarchical structure and representation learning in the document level, we can extend the Hierarchical Structured Model to document-level classification.

References

Bird, S., Klein, E., and Loper, E. Natural language processing with python. 2009.

Chung, J., Gülçehre, Ç., Cho, K., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014. URL http://arxiv.org/abs/1412.3555.

Davidson, T., Bhattacharya, D., and Weber, I. Racial bias in hate speech and abusive language detection datasets. *CoRR*, abs/1905.12516, 2019. URL http://arxiv.org/abs/1905.12516.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019.

Eisner, B., Rocktäschel, T., Augenstein, I., Bosnjak, M., and Riedel, S. emoji2vec: Learning emoji representations from their description. In *SocialNLP@EMNLP*, 2016.

Girshick, R. B., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pp. 580–587. IEEE Computer Society, 2014. doi: 10.1109/CVPR.2014.81. URL https://doi.org/10.1109/CVPR.2014.81.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pp. 655–665. The Association for Computer Linguistics, 2014. doi: 10.3115/v1/p14-1062. URL https://doi.org/10.3115/v1/p14-1062.

Kim, Y. Convolutional neural networks for sentence classification. In Moschitti, A., Pang, B., and Daelemans, W. (eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1746–1751. ACL, 2014. doi: 10.3115/v1/d14-1181. URL https://doi.org/10.3115/v1/d14-1181.

Orignal text	<user> nigga then how can you say bo2 is the best cod 😐 <url>
ID-LSTM	~~<user>~~ **nigga** ~~then~~ **how can** ~~you~~ **say bo2** ~~is the~~ **best cod** 😐 ~~<url>~~
ID-LSTM+a	~~<user>~~ **nigga** ~~then~~ **how can you say bo2** ~~is the~~ **best cod** 😐 ~~<url>~~
ID-BiLSTM	~~<user>~~ **nigga** ~~then how~~ **can you say bo2** ~~is~~ **the best cod** 😐 ~~<url>~~
ID-BiLSTM+a	~~<user>~~ **nigga** ~~then how~~ **can you say bo2** ~~is the~~ **best cod** 😐 ~~<url>~~
BERT+ID-LSTM	~~<user>~~ **nigga then how** ~~can you~~ **say bo2** ~~is the~~ **best cod** 😐 ~~<url>~~
BERT+ID-BiLSTM	~~<user>~~ **nigga then how** ~~can you~~ **say bo2** ~~is the~~ **best cod** 😐 ~~<url>~~
BERT+ID-LSTM	~~<user>~~ **nigga then how** ~~can you~~ **say bo2** ~~is the~~ **best cod** 😐 ~~<url>~~
HS-LSTM	<user> | nigga then how can you say bo2 | is the best cod 😐| <url>
HS-LSTM+a	<user> nigga | then how can you say | bo2 is the best cod |😐<url>
HS-BiLSTM	<user> | nigga then how can you say | bo2 is the best cod 😐| <url>
HS-BiLSTM+a	<user> nigga | then how can you say bo2 is the | best cod |😐 <url>
BERT+HS-LSTM	<user> nigga | then how can you | say bo2 is the best cod 😐| <url>
BERT+HS-Bi	<user> nigga | then how can you say | bo2 is the best cod 😐|<url>

Figure 9. Example tweet after Information Distilled and Hierarchical Structured models

Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *EMNLP*, 2014.

Tai, K. S., Socher, R., and Manning, C. D. Improved semantic representations from tree-structured long short-term memory networks, 2015.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A. J., and Hovy, E. H. Hierarchical attention networks for document classification. In Knight, K., Nenkova, A., and Rambow, O. (eds.), *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pp. 1480–1489. The Association for Computational Linguistics, 2016. doi: 10.18653/v1/n16-1174. URL https://doi.org/10.18653/v1/n16-1174.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., and Kumar, R. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval)*, 2019a.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., and Kumar, R. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of NAACL*, 2019b.

Zhang, T., Huang, M., and Zhao, L. Learning structured representation for text classification via reinforcement learning, 2018a. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16537.

Zhang, T., Huang, M., and Zhao, L. Learning structured representation for text classification via reinforcement learning. In McIlraith, S. A. and Weinberger, K. Q. (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 6053–6060. AAAI Press, 2018b. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16537.