Paper 14 | Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling by Peng Zhou et. al – published in 2016

Abstract

1. RNN can utilise distributed representations of words by first converting the tokens comprising each text into vectors, which form a matrix. This matrix includes two dimensions: **time-step dimension** and **feature vector dimension**
2. Most existing models usually utilise one-dimensional max pooling operation only on the time-step dimension to obtain a fixed-length vector and this is said to have destroy the structure of the feature representation as the features on the feature vector are not mutually independent
3. Therefore, this paper decides to use 2D pooling operation over the two dimensions in the hope to sample more meaningful features for sequence modelling tasks
4. This paper applies 2D max pooling to obtain a fixed-length representation of text and 2D convolution to sample more meaningful information of the matrix
5. Experiments are conducted on six text classification tasks, including:
    - Sentiment analysis
    - Question classification
    - Subjectivity classification
    - Newsgroup classification
6. One of the proposed models achieves highest accuracy on SST binary classification and fine-grained classification tasks

Introduction

1. This paper proposes Bidirectional LSTM with 2D max pooling (BLSTM-2DPooling) to capture features on both the time-step dimension and the feature vector dimension. It first utilises BLSTM to transform text into vectors and then apply 2D max pooling to obtain a fixed-length vector
2. This paper also applies 2D convolution (BLSTM-2DCNN) to capture more meaningful features to represent the input text
3. To better understand the effect of 2D convolution and 2D max pooling, this paper conducts experiments on SST fine-grained task whereby it depicts the performance of the proposed models on different length of sentences and also conducts a sensitivity analysis of 2D filter and max pooling size

Model

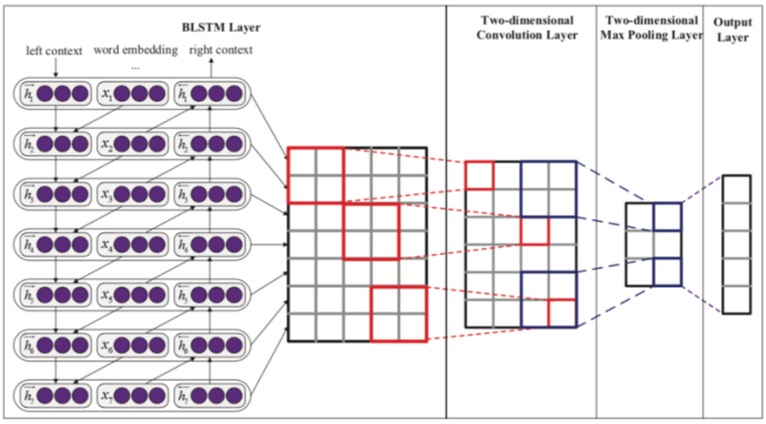

Figure 1: A BLSTM-2DCNN for the seven word input sentence. Word embeddings have size 3, and BLSTM has 5 hidden units. The height and width of convolution filters and max pooling operations are 2, 2 respectively.

1. The overall model consists of 4 parts:
    - BLSTM layer
    - 2D Convolution layer
    - 2D Max-pooling layer
    - Output layer
2. BLSTM
    - BLSTM is utilised to capture the past and the future information. The network contains two sub-networks for the forward and backward sequence context respectively. The output of ith word is the element-wise sum of both the forward and backward pass outputs
3. 2D CNN
    - A convolution operation involves a 2D filter (k x d size) when it is applied to a window of k words and d feature vectors
    - The convolution layer may have multiple filters for the same size filter to learn complementary features, or multiple kinds of filter with different size
4. 2D Max-pooling
    - Utilised to obtain a fixed-length vector
5. Output layer
    - The output of 2D Max-pooling is the whole representation of the input text S
    - The output is passed to a softmax classifier layer to predict the semantic relation label from a discrete set of classes

$$\hat{p}(y|s) = softmax\left(W^{(s)}h^* + b^{(s)}\right)$$
$$\hat{y} = \arg\max_y \hat{p}(y|s)$$

Results

NN	Model	SST-1	SST-2	Subj	TREC	MR	20Ng
ReNN	RNTN (Socher et al., 2013)	45.7	85.4	-	-	-	-
	DRNN (Irsoy and Cardie, 2014)	49.8	86.6	-	-	-	-
CNN	DCNN (Kalchbrenner et al., 2014)	48.5	86.8	-	93.0	-	-
	CNN-non-static (Kim, 2014)	48.0	87.2	93.4	93.6	-	-
	CNN-MC (Kim, 2014)	47.4	88.1	93.2	92	-	-
	TBCNN(Mou et al., 2015)	51.4	87.9	-	96.0	-	-
	Molding-CNN (Lei et al., 2015)	51.2	88.6	-	-	-	-
	CNN-Ana (Zhang and Wallace, 2015)	45.98	85.45	93.66	91.37	81.02	-
	MVCNN (Yin and Schütze, 2016)	49.6	89.4	93.9	-	-	-
RNN	RCNN (Lai et al., 2015)	47.21	-	-	-	-	96.49
	S-LSTM (Zhu et al., 2015)	-	81.9	-	-	-	-
	LSTM (Tai et al., 2015)	46.4	84.9	-	-	-	-
	BLSTM (Tai et al., 2015)	49.1	87.5	-	-	-	-
	Tree-LSTM (Tai et al., 2015)	51.0	88.0	-	-	-	-
	LSTMN (Cheng et al., 2016)	49.3	87.3	-	-	-	-
	Multi-Task (Liu et al., 2016)	49.6	87.9	94.1	-	-	-
Other	PV (Le and Mikolov, 2014)	48.7	87.8	-	-	-	-
	DAN (Iyyer et al., 2015)	48.2	86.8	-	-	-	-
	combine-skip (Kiros et al., 2015)	-	-	93.6	92.2	76.5	-
	AdaSent (Zhao et al., 2015)	-	-	**95.5**	92.4	**83.1**	-
	LSTM-RNN (Le and Zuidema, 2015)	49.9	88.0	-	-	-	-
	C-LSTM (Zhou et al., 2015)	49.2	87.8	-	94.6	-	-
	DSCNN (Zhang et al., 2016)	49.7	89.1	93.2	95.4	81.5	-
ours	BLSTM	49.1	87.6	92.1	93.0	80.0	94.0
	BLSTM-Att	49.8	88.2	93.5	93.8	81.0	94.6
	BLSTM-2DPooling	50.5	88.3	93.7	94.8	81.5	95.5
	BLSTM-2DCNN	**52.4**	**89.5**	94.0	**96.1**	82.3	**96.5**

Table 2: Classification results on several standard benchmarks. **RNTN**: Recursive deep models for semantic compositionality over a sentiment treebank (Socher et al., 2013). **DRNN**: Deep recursive neural networks for compositionality in language (Irsoy and Cardie, 2014). **DCNN**: A convolutional neural network for modeling sentences (Kalchbrenner et al., 2014). **CNN-nonstatic/MC**: Convolutional neural networks for sentence classification (Kim, 2014). **TBCNN**: Discriminative neural sentence modeling by tree-based convolution (Mou et al., 2015). **Molding-CNN**: Molding CNNs for text: non-linear, non-consecutive convolutions (Lei et al., 2015). **CNN-Ana**: A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification (Zhang and Wallace, 2015). **MVCNN**: Multichannel variable-size convolution for sentence classification (Yin and Schütze, 2016). **RCNN**: Recurrent Convolutional Neural Networks for Text Classification (Lai et al., 2015). **S-LSTM**: Long short-term memory over recursive structures (Zhu et al., 2015). **LSTM/BLSTM/Tree-LSTM**: Improved semantic representations from tree-structured long short-term memory networks (Tai et al., 2015). **LSTMN**: Long short-term memory-networks for machine reading (Cheng et al., 2016). **Multi-Task**: Recurrent Neural Network for Text Classification with Multi-Task Learning (Liu et al., 2016). **PV**: Distributed representations of sentences and documents (Le and Mikolov, 2014). **DAN**: Deep unordered composition rivals syntactic methods for text classification (Iyyer et al., 2015). **combine-skip**: skip-thought vectors (Kiros et al., 2015). **AdaSent**: Self-adaptive hierarchical sentence model (Zhao et al., 2015). **LSTM-RNN**: Compositional distributional semantics with long short term memory (Le and Zuidema, 2015). **C-LSTM**: A C-LSTM Neural Network for Text Classification (Zhou et al., 2015). **DSCNN**: Dependency Sensitive Convolutional Neural Networks for Modeling Sentences and Documents (Zhang et al., 2016).

1. This paper implemented 4 models
    - BLSTM
    - BLSTM-Att
    - BLSTM-2DPooling
    - BLSTM-2DCNN
2. BLSTM-2DCNN achieves excellent performance on 4 out of 6 tasks, especially 52.4% and 89.5% test accuracies on SST-1 and SST-2 respectively
3. The paper tests the 4 models on document-level dataset 20Ng to assess whether it is possible to use the proposed models for datasets that have a substantial number of words. Results show that BLSTM-2DCNN achieved a comparable result relative to RCNN
4. Relative to RecNN, the proposed models do not depend on external language-specific features such as dependency parse trees
5. BLSTM-2DCNN is an extension of BLSTM-2DPooling and the results show that the former model can capture more dependencies in text
6. Compared to DSCNN, BLSTM-2DCNN outperforms it on five datasets
7. Sensitivity analysis on SST-1 dataset

- The paper found that both BLSTM-2DPooling and BLSTM-2DCNN outperform the other two models, suggesting that both 2D convolution and 2D max pooling are able to encode semantically-useful structural information
- Accuracies decline with the length of sentences increasing
- In terms of what is the best 2D filter and max pooling size to get better performance, the result shows that the best accuracy (52.6%) was achieved with 2D filter size (5, 5) and 2D max pooling size (5, 5)
- This shows that filter tuning can further improve the performance. A large filter can detect more features but will take up more storage space and consume more time

Conclusion

1. Introduced two combination models
    a. BLSTM-2DPooling
    b. BLSTM-2DCNN (extension of BLSTM-2DPooling)
2. Both models can hold information from the time-step and feature vector dimension
3. The experiments results demonstrate that BLSTM-2DCNN outperforms RecNN, RNN and CNN models as well as BLSTM-2Dpooling and DSCNN
4. BLSTM-2DCNN achieves highest accuracy on SST-1 and SST-2 datasets
5. The sensitivity analysis on SST-1 dataset shows that larger filter can detect more features which may lead to performance improvement