# “Current Topics in Computer Science Seminar”
# A C-LSTM Neural Network for Text Classification

Chunting Zhou & Francis C.M. Lau (Uni. of H.K.)   Chonglin Sun (Dalian Uni. of Tech)   Zhiyuan Liu (Tsinghua Uni, Beijung)

**Paper quality analysis:** This paper was published on website arxiv.org, owned and operated by Cornell University. The paper has 249 citations.

**Presenter:** This scientific research paper: A C-LSTM Neural Network for Text Classification was published on 30th Nov, 2015 by authors Chunting Zhou[1], Chonglin Sun[2], Zhiyuan Liu[3], Francis C.M. Lau[1].
I, Ms. Pallavi Khadse (Group-6) is presenting this paper. Date of submission is on 22nd Sept, 2019.

## I. Introduction

This paper is making its contribution on machine learning of Natural Language Processing (NLP) for sentence representation and text classification.

**Challenges:** The challenges are: 1.) to predict sentiment polarity of movie reviews based on Stanford Sentiment Treebank (SST) and, 2.) classify questions into a specific type based on TREC benchmark.

**Goals/Contributions:** A new architecture called C-LSTM is introduced by combing the advantages of two models, that are 1.) Convolutional Neural Network (CNN) and 2.) Long Short-Term Memory Recurrent Neural Network (LSTM RNN) for sentence modeling. The goal of the proposed model was to achieve excellent results in sentiment analysis and question classification as compared with the other published baseline neural network models.

## II. Method/approach/Architecture/Framework

In this paper, CNN is applied to text data and consecutive window features fed directly to LSTM (please refer Figure 3 in the attached Infographic). This helps C-LSTM architecture to learn long-range dependencies from higher-order sequential features. CNN is built upon word sequences and not on syntactic parse tree. A standard LSTM Recurrent Neural Network (RNN) is adopted as its design is for time-series data for learning long-term dependencies in the sequence of higher-level features.

The entire proposed model was trained by minimizing the cross-entropy error. Given a training sample $x^{(i)}$ and its true label $y^{(i)} \in \{1, 2, \cdot \cdot \cdot, k\}$ where k is the no. of possible labels & the estimated probabilities $y^{\sim(i)} \in [0, 1]$ for each label $j \in \{1, 2, \cdot \cdot \cdot, k\}$, the error is defined as:

$$L(\mathrm{x}^{(i)}, y^{(i)}) = \sum^{k} 1\{y^{(i)} = j\} \log(\widetilde{y}_j^{(i)}) \quad (1)$$

The sentences shorter than the maximum length of the sentence in the training set, were padded for fixed-length input to the convolutional layer. The word vectors are initialized using pre-trained word2vec vectors having dimensionality of 300. Hyperparameters like dropout and L2 weight were employed for regularization.

Datasets used were: 1.) for sentiment classification, 11855 movie reviews from Stanford Sentiment Treebank (SST) benchmark. In this, fine-grained classification & binary classification tasks were considered. And, 2.) for question classification, 5452 labelled questions from TREC benchmark.

## III. Results and Discussion

Results obtained are shown in Table 1 & Table 2 below. On Sentiment analysis, 1.) Comparable results were achieved with respect to those models that heavily rely on linguistic annotations & knowledge, especially syntactic parse trees. This indicates that proposed model C-LSTM will be more feasible for various scenarios, 2.) Result comparison with single LSTM & CNN shows that C-LSTM does learn long-term dependencies across sequences of higher-level representations better.

On Question-type classification, 1.) Results consistently outperforms all (considered) published neural baseline models, meaning that C-LSTM captures intents of TREC question well, 2.) Results were close to Support Vector Model (SVM) that depends on highly human-designed engineered features whereas C-LSTM do not require this as it automatically learns semantic sentence representations.

**Table 1 Results based on Sentiment Analysis**

| Model | Comparison with baseline neural models on SST. | | |
|---|---|---|---|
| | *Fine-grained (%)* | *Binary (%)* | *Reported in* |
| SVM | 40.7 | 79.4 | Socher et al., 2013b |
| CNN-non-static | 48.0 | 87.2 | Kim, 2014 |
| Dependency Tree-LSTM | 48.4 | 85.7 | Tai et al., 2015 |
| Consituency Tree-LSTM | 51.0 | 88.0 | Tai et al., 2015 |
| LSTM | 46.6 | 86.6 | This paper |
| Bi-LSTM | 47.8 | 87.9 | This paper |
| C-LSTM | 49.2 | 87.8 | This paper |

**Table 2 Results based on Question Classification**

| Model | Question classification accuracy on TREC. | |
|---|---|---|
| | *Fine-grained (%)* | *Binary (%)* |
| SVM | 95.0 | Silva et al., 2011 |
| Ada-CNN | 92.4 | Zhao et al., 2015 |
| CNN-non-static | 93.6 | Kim 2014 |
| LSTM | 93.2 | This paper |
| Bi-LSTM | 93.0 | This paper |
| C-LSTM | 94.6 | This paper |

## IV. References

[1] Sepp Hochreiter and J¨urgen Schmidhuber. 1997. Long short-term memory. Neural computation, 9(8):1735–1780.

[2] Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. Association for Computational Linguistics (ACL).

[3] [Kim2014] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In Proceedings of Empirical Methods on Natural Language Processing.

[4] Tara N Sainath, Oriol Vinyals, Andrew Senior, and Hasim Sak. 2015. Convolutional, long short-term memory, fully connected deep neural networks. IEEE International Conference on Acoustics, Speech and Signal Processing.

[5] Duyu Tang, Bing Qin, and Ting Liu. 2015. Documentmodeling with gated recurrent neural network for sentiment classification. In Proceedings of Empirical Methods on Natural Language Processing.

# A C-LSTM Neural Network for Text Classification

Published on 30th Nov, 2015 by Chunting Zhou[1], Chonglin Sun[2], Zhiyauan Liu[3], Francis C.M. Lau[1]

….PRESENTED BY MS. PALLAVI KHADSE (GR-6)

*introduction*:

this paper is the report of the experiment carried with combing two architectures - convolutional neural network (cnn) & long-short term memory recurrent neural network (lstm rnn) for sentence representation & text classification. The unified & novel model so obtained is called c-lstm. This proposed model is evaluated on two tasks – sentiment classification and question type classification. Through the experiment it was learned that c-lstm architecture outperfoms both cnn and lstm and can achieve excellent performance on sentiment analysis & question classification tasks.

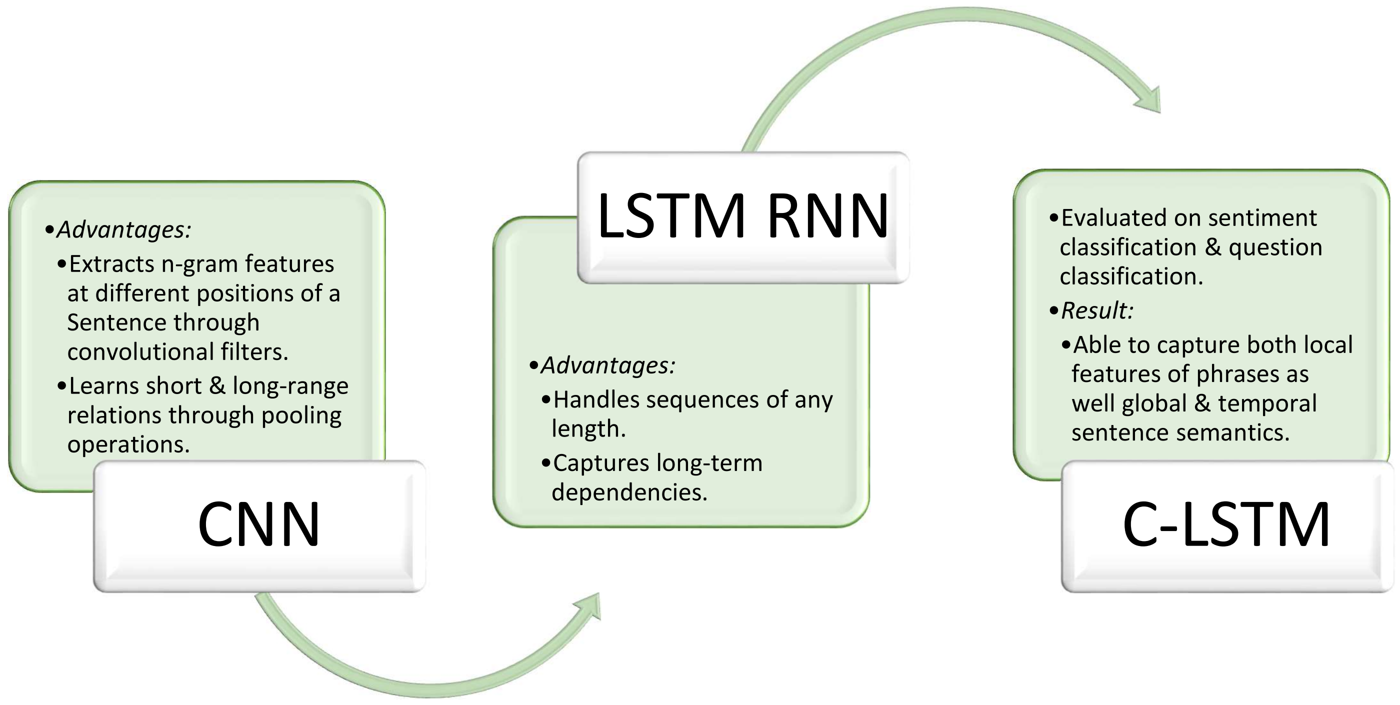


Figure 1: CNN &LSTM RNN combined to get a new architecture C-LSTM.

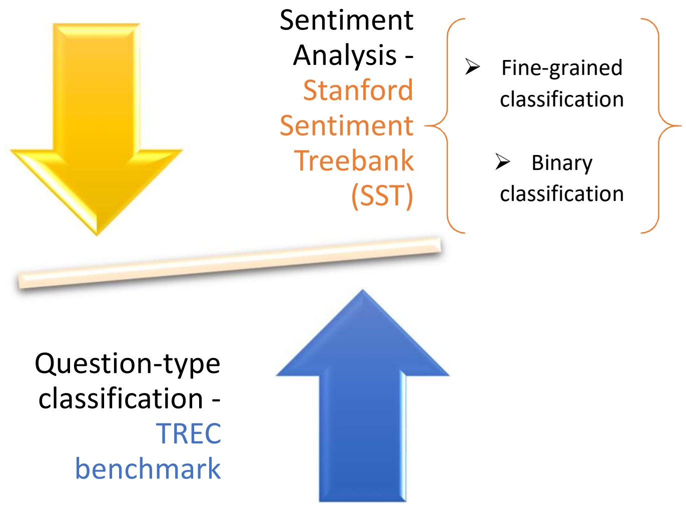


Figure 2: The C-LSTM evaluated based on above tasks.

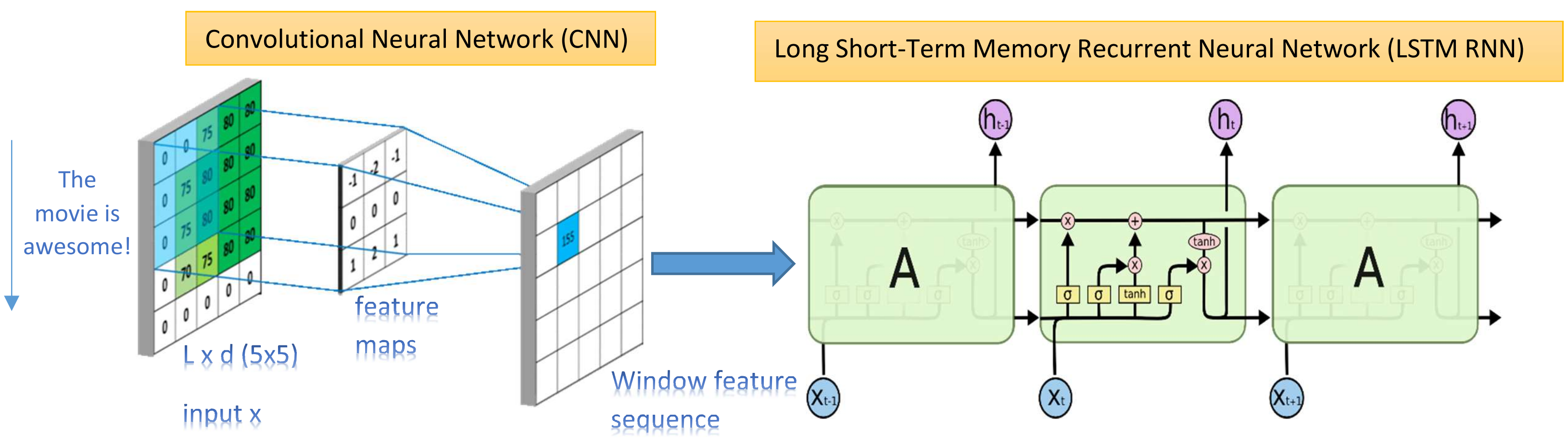


Figure 3: CNN applied to text data and consecutive window features fed directly to LSTM RNN.

LSTM transition function:

$i_t = \sigma(W_t \cdot [h_{t-1}, x_t] + b_t)$
$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$
$q_t = tanh(W_q \cdot [h_{t-1}, x_t] + b_q)$
$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$
$c_t = f_t \odot c_{t-1} + i_t \odot q_t$
$h_t = o_t \odot tanh(ct)$

where,
σ - the sigmoid function,
⊙ - element-wise multiplication,
tanh - the hyperbolic tangent function,
ft - the function to control to what extent the information from the old memory cell is going to be thrown away,
$i_t$ – to control how much new information is going to be stored in the current memory cell,
$o_t$ - to control what to output based on the memory cell $c_t$.

## *LEARNING C-LSTM FOR TEXT CLASSIFICATION:*

1. Padding applied to each sentence that has shorter length than the maximum length of the sentence in the training set.
2. Word2vec vectors used for initializing word vectors.
3. For regularization, dropout and L2 weight regularizations used.

## *EXPERIMENT:*

Experimental Settings:

- The model implemented based on Theano, a python library.

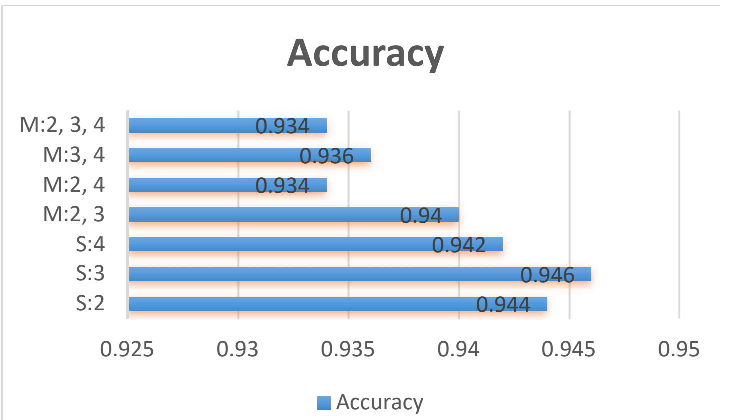


Figure 4: Prediction accuracies on TREC questions with different filter size strategies. For the vertical axis, S means single convolutional layer with the same filter length, and M means multiple convolutional layers in parallel with different filter lengths.

## *RESULTS & CONCLUSION:*

**On Sentiment Classification**:

- Comparable results achieved with respect to the state-of-the-art models that heavily rely on syntactic parse trees. This indicates C-LSTM will be more feasible for various scenarios.
- LSTM found better in learning long-term dependencies across sequences of higher-level representation.

**On Question Type Classification:**

- Result outperformed all published neural baseline models. Means C-LSTM captures intentions of TREC questions well.
- The result were close to the state-of-the-art Support Vector Machine (SVM) model. C-LSTM does not require any human-designed features as required by SVM because it has an ability of automatically learning semantic sentence representations.

*References*: [Bastien et al.2012], [Cho et al.2014], [Collobert et al.2011], [Denil et al.2014], [Devlin et al.2014], [Hinton et al.2012], [Hochreiter and Schmidhuber1997], [Irsoy and Cardie2014], [Johnson and Zhang2015], [Kalchbrenner et al.2014], [Kim2014], [Le and Mikolov2014], [Lei et al.2015], [Li and Roth2002], [Li et al.2015], [Mikolov et al.2013b], [Mou et al.2015], [Nair and Hinton2010], [Pascanu et al.2014], [Sainath et al.2015], [Silva et al.2011], [Socher et al.2012], [Socher et al.2013a], [Socher et al.2013b], [Sutskever et al.2014], [Tai et al.2015], [Tang et al.2015], [Tieleman and Hinton2012], [Xu et al.2015], [Zhao et al.2015].