http://www.ftsm.ukm.my/apjitm
Asia-Pacific Journal of Information Technology and Multimedia
Jurnal Teknologi Maklumat dan Multimedia Asia-Pasifik
Vol. x No. x, December 2017: xx- xx
e-ISSN: 2289-2192

ATTENTIONAL CNN-LSTM FOR TARGET-DEPENDENT SENTIMENT CLASSIFICATION OF ON-LINE REVIEW

Z. Madhoushi, S. Hejazi[1], A. R. Hamdan, S. Zainudin
Center for Artificial Intelligence Technology, Faculty of Information Science and Technology,
Universiti Kebangsaan Malaysia 43600 Bangi, Selangor Darul Ehsan, Malaysia.
[1]Arian University, Iran

ABSTRACT

Target-dependent sentiment classification (TDSC) aims to predict the sentiment polarity of sentences towards specific aspects of an item or product. Classical successful supervised models usually use Support Vector Machine (SVM) or Conditional random field (CRF) with selected features. With the advance of deep learning many models created for this task using models such as Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM). In this paper we use Combination of CNN and LSTM for a hybrid framework to exploit both spatial and temporal information of sentence for TDSC. We compare the outcomes of the LSTM and CNN alone with the CNN-LSTM combined system in order to investigate how spatial information affects the performances. We also take advantage of attention mechanism to use aspect information during training. This work build a new framework for TDSC. It also improve accuracy of state of the art deep learning methods for this task.

Keywords: *Convolutional Neural Network, Long Short Term Memory, Target-dependent sentiment classification.*

1 Introduction

TDSC basic task is to extract people's opinions expressed on aspects of entities. For example, in the sentence "I *brought a Sony camera yesterday, and its picture quality is great*" the TDSC system should identify the author expressed a positive opinion about the '*picture quality'* of the Sony camera. Here '*picture quality'* is an aspect and Sony camera is the entity. Aspects are attributes or components of entity (e.g.,'LCD ', 'battery life', etc. for a digital camera) and ratings/polarity are the intended interpretation of user

satisfaction in terms of numerical values or simply positivity and negativity of the sentiment (Liu, 2012).

Deep learning architectures such as LSTM (Ruder, Ghaffari, & Breslin, 2016), CNN (H. Wu, Gu, Sun, & Gu, 2016) have recently become popular for sentiment analysis. LSTM is useful for extracting long temporal information and CNN for static spatial information (Z. Wu, Wang, Jiang, Ye, & Xue, 2015). CNN known as hierarchical feature extractor while LSTM uses selective memory of historical information to create features in a sequence.

Also attention mechanism is an interesting approach for TDSC that can concentrate on different parts of a sentence for different aspects. Yang, Tu, Wang, Xu, & Chen, 2017 present two types of TDSC to learn the alignment between the targets and the most distinguishing features in a sentence. To the best of our knowledge there is no work in the literature that investigate how spatial and temporal information affects the performances of TDSC. In this paper we use Combination of CNN and LSTM with attention for a hybrid framework to exploit both spatial and temporal information of sentence for TDSC.

2 Related work

Most of the early works on TDSC use SVM (Support Vector Machine), (Blair-Goldensohn & Hannan, 2008), (Y. Wu, Zhang, Huang, & Wu, 2009) (Jiang, Yu, Zhou, Liu, & Zhao, 2011) with hand engineered features such as n-grams, negation words, and sentiment lexicons. Sequential learning (or sequential labelling) such as HMM (Jin, Ho, & Srihari, 2009) and CRF (Shariaty & Moghaddam, 2011) , (Choi & Cardie, 2010), (Li et al., 2010) are other approaches in the literature.

Classifiers uses manually engineered features which is labor intensive and cannot create a true representation for sentences. Deep learning automatically learns latent features as distributed vectors and have recently been shown to outperform many machine learning methods on this task. (Liu, Joty, & Meng, 2015) uses two types Long Short term Memory Recurrent Neural Network (LSTM RNN) with many word vector training. There are also some recent hybrid deep learning models such as (W. Wang, Pan, Dahlmeier, & Xiao, 2016) which integrate RNN with CRF and (Du et al., 2016) which integrate CNN with LDA.

(Tang, Qin, Feng, & Liu, 2015) presents two types of TC-LSTM, TD-LSTM which considers the aspect information during training for TDSC. Attention mechanism is introduced by (Luong, Pham, & Manning, 2015). (Wang, Huang, Zhao, & Zhu, 2014) uses attention that can capture the key part of sentence for a given aspect. (Yang et al., 2017) presents two types of attention that learns to assign attention scores to different word locations according to their relevance to the task. In this work we present an attentional deep neural network architecture that takes advantage of the CNN and LSTM and joint them together for TDSC.

3 Model

3.1 Word Embedding

Word embedding are vector representation for word. The commonly used are random initialization and unsupervised pre-training of word embedding. In our experiment, we used unsupervised pre-training of the *word2vec* (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013) method. We then fine-tune the word vectors along with other model parameters during training.

3.2 CNN

Based on (Kim, 2014) the one-dimensional convolution involves a filter vector sliding over a sequence and detecting features at different positions. Let $x_i \in R^d$ be the d -dimensional word vectors for the I -th word in a sentence. Let $x \in R_{L \times d}$ be the input sentence where L is the length of the sentence. Let k be the length of the filter, and the vector $m \in R_{k \times d}$ is a filter for the convolution operation. For each position j in the sentence, we have a window vector w_j with k consecutive word vectors which is:

$$w_j = [x_j, x_{j+1}, \cdots, x_{j+k-1}] \qquad (1)$$

A filter m convolves with the window vectors (k -grams) at each position in a valid way to generate a feature map $c \in R^{L-k+1}$; each element c_j of the feature map for window vector w_j is produced as follows:

$c_j = f(w_j \circ m + b)$, (2) where $\circ$ is element-wise multiplication, $b \in R$ is a bias term and f is a nonlinear transformation function that can be sigmoid, hyperbolic tangent, etc. In this study, we choose ReLU (Nair & Hinton, 2010) as the nonlinear function. Then, we apply a pairwise max pooling operation over the feature maps to capture the most important features of P_1, P2 and P3. These features are higher-level sequences of word (sentence) features. Figure 1 illustrates the CNN architecture.

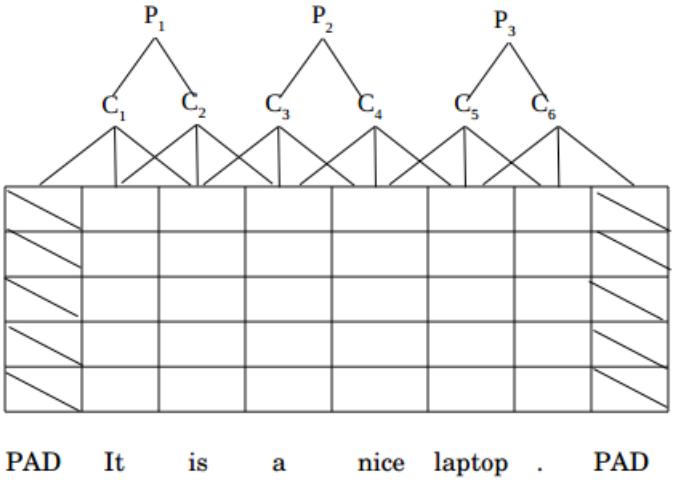

FIGURE 1. CNN architecture

3.3 LSTM

Long Short Term Memory networks (LSTM) (Hochreiter & Schmidhuber, 1997) are a special kind of recurrent nets that are capable of learning long-term dependencies. So they designed to recognize patterns in sequences of data, such as text. LSTM controls information outside the normal flow of the recurrent network with the help of gates. In the standard LSTM architecture, there are three gates and a cell memory state. Figure 2 shows the architecture of a standard LSTM.

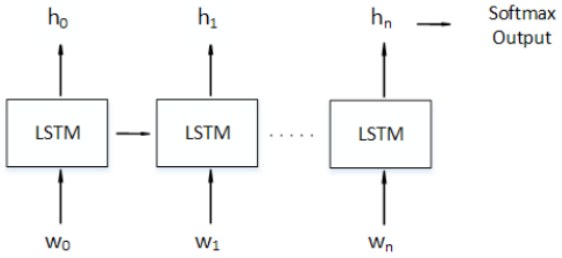

Figure 2-The architecture of a standard LSTM.
$w_1, w_2 \ldots, w_n$ represent word vectors in a sentence with length n. $h_1, h_2, \ldots, h_n$ are the hidden vectors.

The inputs for each cell is:

$$X = \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} \quad (3)$$

x_t includes the inputs of LSTM cell unit, representing the word embedding vectors w_t in Figure 2. The vector of hidden layer is h_t. In each LSTM cell there are three gates. f, i and o are forget gate, input gate and output gate respectively. These gates can be computed as follows:

$$f_t = \sigma(W_f . X + b_f) \quad (4)$$

$$i_t = \sigma(W_i . X + b_i) \quad (5)$$

$$o_t = \sigma(W_o . X + b_o) \quad (6)$$

Each cell is computed as follow:

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_c . X + b_c) \quad (7)$$

Each hidden unit is computed as follow:

$$h_t = o_t \odot tanh(c_t) \quad (8)$$

W_i, W_f ,$W_o \in R^{d \times 2d}$ are the weighted matrices and b_i, b_f , $b_o \in R^d$ are biases of LSTM to be learned during training. σ is the sigmoid function and ⊙ stands for element-wise multiplication. Both h and c are initialized with zeros.

We regard the last hidden vector h_n as the representation of sentence and put h_n into a softmax layer after linearizing it into a vector whose length is equal to the number of class labels. In our work, the final output of classifier is $O \subseteq R^{1 \times 4}$ which classifies the sentence as three categories of *positive*, *negative*, *neutral*.

Bidirectional LSTM (BLSTM) first used in (Graves, Fernández, & Schmidhuber, 2005) is an LSTM that process the data in both left and right directions and creates two separate hidden layers, which are then feed forwards to the same output layer.

3.3.1 Attentional LSTM

Adding attention layer to LSTM helps the network to capture the key part of sentence for a given aspect. Based on (Y. Wang, Huang, Zhao, & Zhu, 2014) the attention mechanism will produce an attention weight vector α and a weighted hidden representation r. Let $H \in R_{d\times N}$ be a matrix consisting of hidden vectors $[h_1, \ldots, h_N]$ that the LSTM produced, where d is the size of hidden layers and N is the length of the given sentence. Furthermore, v_a represents the embedding of aspect and $eN \in RN$ is a vector of 1s. r is computed as follow:

$$M = tanh\left[\begin{matrix} W_h * H \\ W_v * v_a \otimes e_N \end{matrix}\right] \quad (9)$$

$$\alpha = \text{softmax}(w^T M) \quad (10)$$

$$r = H\alpha^T \quad (11)$$

where, $M \in R_{(d+da)}\times N$, $\alpha \in R^N$, $r \in R_d$. $W_h \in R_{d\times d}$, $W_v \in R_{da\times da}$ and $w \in R_{d+da}$ are projection parameters. α is a vector consisting of attention weights and r is a weighted representation of sentence with given aspect. The operator in 9 (a circle with a multiplication sign inside) means: $v_a \otimes e_N = [v; v; \ldots; v]$, that is, the operator repeatedly concatenates v for N times, where e_N is a column vector with N 1s. $W_v v_a \otimes e_N$ is repeating the linearly transformed v_a as many times as there are words in sentence. The final sentence representation is given by:

$$h^* = \tanh(W_p r + W_x h_N) \quad (12)$$

Where, $h^* \in R_d$, W_p and W_x are projection parameters to be learned during training. Figure 3 illustrates the Attentional-BLSTM architecture.

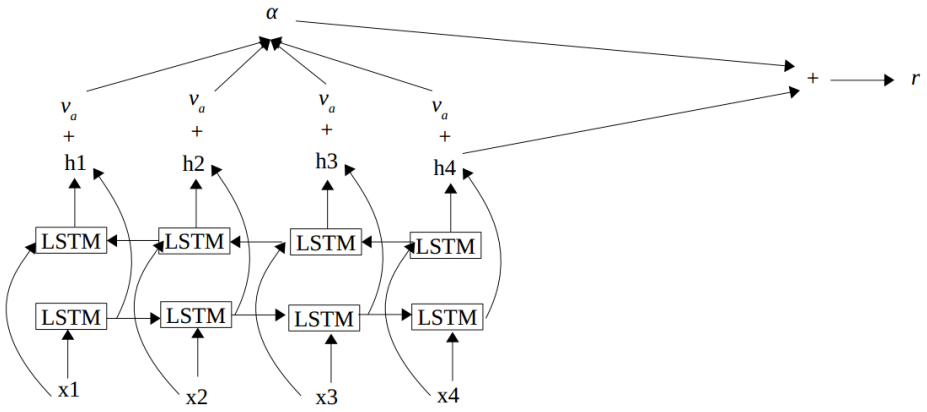

Figure 3- Attentional-BLSTM architecture

3.4 Attentional CNN-BLSTM

We combine CNN with attentional LSTM to create Attentional CNN-LSTM, and we name it AC-LSTM. In this model CNN extracts higher-level sequences of word features (P_1, P_2, P_3) and LSTM captures long term dependencies of the features extracted by CNN (h_2, h_3, h_4). Considering that LSTM is specified for sequence input, note that average pooling will not break such sequence. The model architecture is illustrated in figure 4.

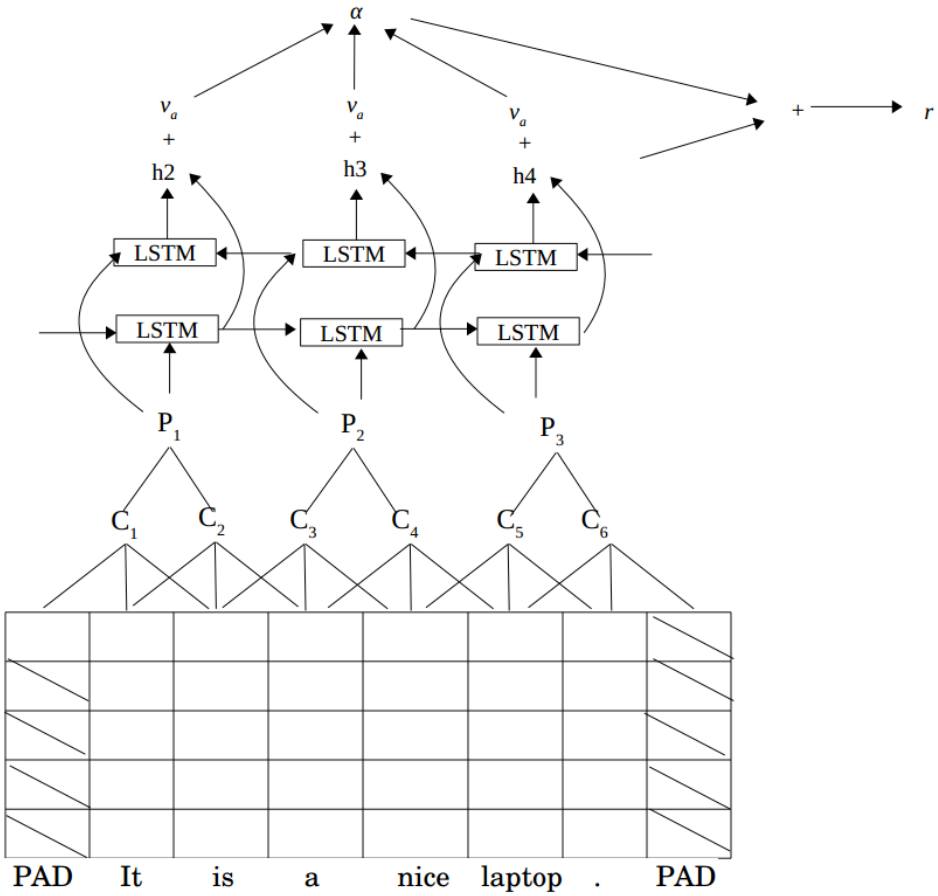

Figure 4- Attentional CNN-LSTM architecture

4 **Experiments**

4.1 *Dataset*

We experiment on the dataset of SemEval 2015 (Pontiki, Galanis, Papageorgiou, Manandhar, & Androutsopoulos, 2015) The dataset consists of customers reviews about Laptop. Each review contains a list of aspects and corresponding polarities. Given a set of predefined aspects, and identified aspects for each sentence, our aim is to find the polarity of each aspect in a sentence.

4.2 *Baseline*

We compare our approach with several baseline methods, including BLSTM (Graves, Fernández, & Schmidhuber, 2005), CNN (Kim, 2014) and the Best result of SemEval 2015 for this task (Pontiki, Galanis, Papageorgiou, Manandhar, & Androutsopoulos,

2015). The best accuracy for this dataset were achieved by Sentiue with a MaxEnt classifier along with features based on n-grams, POS tagging, lemmatization, negation words and publicly available sentiment lexica (MPQA, Bing Liu's lexicon, AFINN). The simple BLSTM and CNN cannot capture any aspect information in sentence, and it gets the sentence level polarity. No aspect information can be used in these models. Therefore we compare our model with Bidirectional form of target dependent LSTM which is called TD-LSTM in (Tang, Qin, Feng, & Liu, 2015) and name it as TD-BLSTM. We also used attention mechanism used in (Y. Wang et al., 2014) and compared the results with A-BLSTM.

To check the effect of combining CNN and BLSTM, we also compare the results with our CNN_BLSTM.

4.3 *Implementation*

We implement our model based on Tensorflow using Keras. Tensorflow is a python library, which supports efficient symbolic differentiation. To take advantage of the efficiency of parallel computation, we train the model on a GPU. For the preprocessing, we just put sentences on separate lines.

We use 300-dimensional word embedding of pre-trained Word2Vec (Mikolov et al., 2013) on Google News to feed the CNN. Our BLSTM have one layer and an output size of 128 dimensions.. We use dropout of 0.5 after BLSTM and l_2 regularization. We train our model to minimize the cross-entropy loss, using the Adam update rule (Kingma & Ba, 2014) and mini-batches of size 32.

5 **Result**

In this experiment, the results are evaluated using classification accuracy and F1-score. Table 1 shows the experiment results on SemEval-2015 dataset. We summarize the experiment results in Table 1.

Table 1- Evaluation results

Model	Accuracy%	F1-score%
Best of SemEval-2015 task 12	79.34	-------
CNN	81.1	81.0
BLSTM	82.4	82.3
TD-BLSTM	82.7	82.6
A-BLSTM	83.0	82.9
Our CNN_BLSTM	82.5	82.4
Our AC-BLSTM	**83.7**	**83.6**

The result verifies the effectiveness of combining temporal and spatial information of a sentence for TDSC. Using attentional mechanism all models achieve better results compare to those without attention which confirms previous finding. For example, the F1-score of A-BLSTM is 0.6% higher than BLSTM.

6 **Conclusion**

In this paper, we build a new framework for TDSC using both CNN and LSTM architecture. We also took advantage of attentional mechanism. The goal was to investigate how spatial and temporal information affects the performances of TDSC. Experiments show that the complex AC-BLSTM model obtain better performance compare with both CNN and LSTM alone. The results also confirm the effectiveness of attentional mechanism which is presented in other baselines.

References

Blair-Goldensohn, S., & Hannan, K. (2008). Building a sentiment summarizer for local service reviews. *WWW Workshop on*. Retrieved from http://www.academia.edu/download/11179325/paper3.pdf

Choi, Y., & Cardie, C. (2010). Hierarchical sequential learning for extracting opinions and their attributes. *Proceedings of the ACL 2010 Conference Short*. Retrieved from http://dl.acm.org/citation.cfm?id=1858892

Graves, A., Fernández, S., & Schmidhuber, J. (2005). Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition. *Artificial Neural Networks: Formal Models and Their Applications–ICANN*, 799–804. Retrieved from ftp://ftp.idsia.ch/pub/juergen/icann2005graves.pdf

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Jin, W., Ho, H., & Srihari, R. (2009). OpinionMiner: a novel machine learning system for web opinion mining and extraction. *Proceedings of the 15th ACM SIGKDD*. Retrieved from http://dl.acm.org/citation.cfm?id=1557148

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751. Retrieved from http://www.aclweb.org/anthology/D14-1181

Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. Retrieved from http://arxiv.org/abs/1412.6980

Li, F., Han, C., Huang, M., Zhu, X., Xia, Y., & Zhang, S. (2010). Structure-aware review mining and summarization. *Proceedings of the 23rd*. Retrieved from http://dl.acm.org/citation.cfm?id=1873855

Liu, B. (2012). Sentiment Analysis and Opinion Mining. *Sentiment Analysis and Opinion Mining*. Retrieved from https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective Approaches to Attention-based Neural Machine Translation. *Emnlp*, (September), 11. https://doi.org/10.18653/v1/D15-1166

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 26* (pp. 3111–3119). Curran

Associates, Inc. Retrieved from http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

Nair, V., & Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. *Proceedings of the 27 Th International Confer- Ence on Machine Learning*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.165.6419&rep=rep1&type=pdf

Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., & Androutsopoulos, I. (2015). SemEval-2015 Task 12: Aspect Based Sentiment Analysis, 486–495. Retrieved from http://www.aclweb.org/anthology/S15-2082

Ruder, S., Ghaffari, P., & Breslin, J. G. (2016). A Hierarchical Model of Reviews for Aspect-based Sentiment Analysis.

Shariaty, S., & Moghaddam, S. (2011). Fine-grained opinion mining using conditional random fields. *Data Mining Workshops (ICDMW),*. Retrieved from http://ieeexplore.ieee.org/abstract/document/6137368/

Tang, D., Qin, B., Feng, X., & Liu, T. (2015). Effective LSTMs for Target-Dependent Sentiment Classification. Retrieved from http://arxiv.org/abs/1512.01100

Wang, Y., Huang, M., Zhao, L., & Zhu, X. (2014). Attention-based LSTM for Aspect-level Sentiment Classification. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 606–615.

Wu, H., Gu, Y., Sun, S., & Gu, X. (2016). Aspect-based Opinion Summarization with Convolutional Neural Networks. *Proceedings of the International Joint Conference on Neural Networks*, *2016–Octob*, 3157–3163. https://doi.org/10.1109/IJCNN.2016.7727602

Wu, Y., Zhang, Q., Huang, X., & Wu, L. (2009). Phrase dependency parsing for opinion mining. *Of the 2009 Conference on Empirical* …. Retrieved from http://dl.acm.org/citation.cfm?id=1699700

Wu, Z., Wang, X., Jiang, Y.-G., Ye, H., & Xue, X. (2015). Modeling Spatial-Temporal Clues in a Hybrid Deep Learning Framework for Video Classification. Retrieved from https://arxiv.org/pdf/1504.01561.pdf

Yang, M., Tu, W., Wang, J., Xu, F., & Chen, X. (2017). Attention-Based LSTM for Target-Dependent Sentiment Classificatio, 5013–5014. https://doi.org/10.1146/annurev.neuro.26.041002.131047