Sentiment Prediction of IMDb Movie Reviews Using CNN-LSTM Approach

1st Mahesh Mishra
dept. of Electronics and Telecommunication
A. C. Patil College of Engineering (Mumbai University)
Kharghar, India
mahesh070901@gmail.com

2nd Amol Patil
dept. of Electronics and Telecommunication
A. C. Patil College of Engineering (Mumbai University)
Kharghar, India
agpatil@acpce.ac.in

2023 International Conference on Control, Communication and Computing (ICCC) | 979-8-3503-3412-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCC57789.2023.10165155

***Abstract*—This article describes sentiment classification of movie reviews given by the user using deep neural networks. Long short term memory (LSTM) and convolutional neural network (CNN) are two popular deep neural networks used for sentiment analysis. Sentiment analysis is carried out on internet movie dataset (IMDb) which consist of 50K movie reviews. CNN and LSTM architectures are used individually and later combination of CNN-LSTM architecture is used. Accuracy and loss metrics measures are plotted for each architecture where LSTM architecture outperforms compare to CNN and CNN-LSTM architecture. Accuracy of GRU, CNN, LSTM and CNN-LSTM architecures are 53% 85%, 87% and 85% respectively. Adam optimizer and binary cross entropy is used for loss function.CNN-LSTM model is very good for long term dependency and accuracy is also good. Combination of CNN-LSTM reduces a training time for larger dataset and CNN has a convolutional layer to extract information by a larger piece of text.**

***Index Terms*—Sentiment analysis, IMDb movie reviews, CNN, LSTM, CNN-LSTM**

I. INTRODUCTION

sentiment analysis is famous in Natural language processing(NLP) and text mining. Because the majority of a product's success depends on the internet reviews it receives, it is currently one of the most crucial and fascinating research areas. Sentiment analysis enables us to comprehend how natural language and human emotions or judgement interact. Reviewing a person's viewpoint on something that is very important to the person who produced it is helpful to us. For instance, in this day and age, no one watches a movie unless they have read positive reviews of it online or from reviewers. The situation is identical the previous era Different machine learning approaches, like Naive Bayes and SVM [1], were employed in the past to analyse sentiment, with good results. .

Due to the increase amount of data, Deep Neural Network architectures have recently demonstrated notable improvements in NLP tasks. Because of its compositions and local in variance, the convolutional neural network (CNN) is one of the most widely used neural network architectures for image categorization. A catchphrase that is needed for sentiment analysis can be easily found in natural text according to CNN's innovative identification technique, which has also demonstrated substantial performance in natural language processing the earlier era Various machine learning techniques, including Naive Bayes and SVM [1], have been used in the past with successful results to analyse sentiment.

Deep Neural Network designs have lately shown appreciable advances in NLP tasks as a result of the expanding amount of information.Due to its compositionality and local invariance, the convolutional neural network (CNN) is one of the most often used neural network architectures for picture categorization. CNN's novel identification method, which has also demonstrated noteworthy proficiency in natural language processing, makes it simple to locate a catchphrase that is required for sentiment analysis in natural text for sentence categorization [3] has also used various Convolution Neural Network variations. CNN has fewer connections, which has the benefit of reducing training time.

Another type of neural network called a recurrent neural network (RNN) is capable of accurately simulating the structural dependence of brief texts or sentences. However, due to its vanishing gradient issue, it is unable to describe long-term dependency [4]. This means that it is unable to hold long-term dependencies and, as a result, cannot accurately represent sentence structure Because of [4] has suggested LSTM, an enhanced variant of Vanilla RNN that offers long-term dependence. The usage of LSTM is widespread in various NLP tasks that rely heavily on the structural dependency of task translation, picture captioning, question answering, and prediction of the next word in a sentence [5] recently shown how well LSTM performs in text sentiment analysis. Since training LSTM takes a lot longer than training CNN architecture, we chose the most straightforward LSTM-CNN architecture. To extract the features for sentiment categorization, we employed this LSTM-CNN. The sentences in the review were first converted into a vector space using the word embedding method described in this paper. Three distinct neural network architectures were used to feed the collected features into a multi-layer perceptron network for the categorization of positive and negative sentiments: LSTM-CNN, CNN, and LSTM. The purpose of the paper is to identify the optimal Deep Neural Network topologies that make seperation outcomes on the IMDb frrdback dataset. Neural network framework Keras [6], a high-level API built, to develop our architectures.
Sentiment analysis plays a crucial role of determining

what customers views on brand,products,real world problems,business,virtual entity. Sentiment analysis aids in the study of user evaluations of a subject so that conclusions can be drawn based on the sentiments from the reviews.

The remaining sections of the article are structured as follows. **Section-II** describes related work done on sentimental analysis. The structures utilised in this paper is reviewed in **Section-III**. The experimental results are explained in **Section-IV**. Finally, **Section-V** described conclusions based on our experimental study

II. Related Work

A. Lexicon-Based Methods

Lexicon-based approaches make the supposition that the effective ratings of a text can be calculated from the sum of the effective scores of its individual words.One can determine the effective scores of a text using various composition techniques based on the effective scores of individual words. Arithmetic mean is a practical composing method. In other words, the average VA values of each word in this sentence

B. Regression-Based Methods

Both at the word level and the sentence level, regression-based techniques for VA prediction have seen extensive study. Wei et al. [10] transferred VA ratings English to Chinese terms at the word level using linear regression. For the purpose of VA prediction, Malandrakis et. al. [11] combined word similarity using a kernel function. A weighted graph approach was employed by Yu et al. [12] and Wang et al. [13] to iteratively determine the VA ratings of emotive words.
proposed using a linear regression model on an phrase level lexicon at the phrase level to forecast the final sentiment score. When calculating sentiment ratings from an affective lexicon the candidate texts are broken down into their individual words. To eliminate terms that are not included in the lexicon, a list of stop words is included. The text's sentiments scores as well as the average word scores are then used to build a regression model.
A sentence or document's valence and arousal levels were predicted by Paltoglou and Thelwall [14] using an five level scale that ranged from less to big value. The sentiment prediction issue was examined from both a classification and regression perspective by the authors. Based on BoW features, both approaches are used. Then, for classification and regression, Support Vector Machine (SVM) and Support Vector Regression $(\epsilon - SVR)$ is utilised. Their experimental findings also demonstrate that regression methods typically result in smaller magnitude mistakes.

III. METHODOLOGY

A. Dataset Description

One of the largest IMDb movie review datasets is this one [7]. There are 50,000 movie reviews in the dataset, split into two categories: favourable and negative. This dataset, where every feedback is encrypt as a series word indexe, can be found at Keras [6]. The dataset has then been divided into training and testing data in a $80:20$ ratio. A validation dataset was created using 20 percent of the training samples. In order to make it simple for the architectures to be trained, we zero-padded the shorter reviews to make all of the reviews the same length.

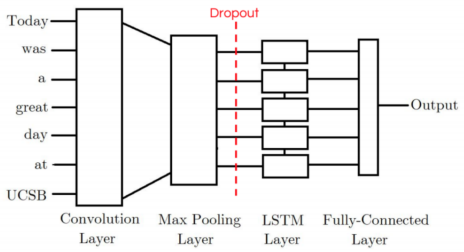

Fig. 1: Accuracy of LSTM

B. Word Embedding

One of the key requirements for the majority of NLP jobs involve natural text is word embedding. To fit any architecture, every word to be transformed to n dimensional. There are 2 ways to go about doing this now. Word embedding and one-hot key encoding using the Bag of Words model are two of them. Bag of words is a relatively sparse form that wastes memory while word embedding provides a close packed presentation of every term. Word embedding is carried out in such a way that related word types are collected together in the n-th dimension. The vector dimension of each embedded word is shown here by the number n. There are two word embedding models that are pre-trained. Word2vec [2] and Glove [8] are two examples. Although each word in our vocabulary of 8000 distinct words is comes in a 100 dimensioal vector, we employed an embedding layer provided by Keas. model is trained using a data of IMDb movies.

C. Convolutional Neural Network

Three layers make up the bulk of the convolutional neural network [2].One mixes the filters and input matrix in the convolution layer. The kernel, often referred to as the filter, is utilised to identify a certain characteristic. In a CNN, numerous filters are employed. Glorot-uniform distribution serves as the initialization point for the filter's weights. Later, through network training, It has been weighted to detect a particular trait. The second layer, referred to as the pooling layer, is where the last neuron layer of one layer is merged with that of a single neuron at the following layer. There are various kinds of pooling, including Maximum Pooling and Average Pooling. Max pooling is employed in the majority of situations. Translation invariance can be provided through pooling. Between two convolution layers, a layer of pooling is applied. Convolution and pooling layers are employed before the third, which is a fully connected layer. The layer's job is to draw conclusions at a higher level from the low characteristic found in the Convolution layer,the CNN receives 500 words each review as input, for 100-dimensional vector for each word the embedding layer it is gone through.

2 convolutional layers were utilised to get characteristics and 2 pooling layers were used. In the convolutional layer, we employed the activation function of ReLU [9].The output of the above 2 layers layers is given to a into a fully-connected layer for classification, then feeds the hidden layer with the features that were extracted. Given that it is a binary classification problem, there is just one sigmoid activation function node in the output layer.

D. Long short Term Memory

LSTM is a new version of Recurrent neural network. The key difference between LSTM and RNN is in LSTM the memory cell that may store or update information based on input present. LSTM have forget gate,input gate and last output gate. The Forget Gate layer, a Sigmoid layer that determines what data must be removed from the memory cell. The second layer, the input gate layer, decides what fresh data will be stored in the memory cell. It is divided further into two layers: a Sigmoid layer and a Tanh layer. The output of the associated LSTM Cell is decided by the third layer, the output gate layer. The gate can be defined mathematically by

$$f_t = \sigma_g(W_f \times x_t + U_f \times h_{t-1} + b_f)$$
$$i_t = \sigma_g(W_i \times x_t + U_i \times h_{t-1} + b_i)$$
$$o_t = \sigma_g(W_o \times x_t + U_o \times h_{t-1} + b_o)$$
$$c^{'} = \sigma_c(W_c \times x_t + U_c \times h_{t-1} + b_c)$$
$$h_t = o_t.\sigma(ct)$$
$$c_t = f_t \times c_{t-1} + i_t.c_t$$

The forget, input, and output gates are represented here by f_t, i_t, and O_t. Layers bias factors are represented by letters bf and b_o. The current unit's input and output are represented by x_t and h_t, respectively, while h_{t-1} is the result of x_t's predecessor, x_{t-1}. tanh and sigmoid layers are represented by tanh and sigmoid layers, respectively. The given LSTM network receives the same input as CNN. To reduce the complexity of the model only one LSTM layer is employed in LSTM network following the Embedding Layer. Similar to this an MLP network was supplied with the LSTM-extracted features for classification. The output layer is identical to CNN network

E. CNN-LSTM

Comparing the dimensional approach to categorical methods that focus on sentiment categorization, suppose a binary classification, can result in a more accurate sentiment analysis (i.e., positive and negative). This article presents a tree-structured regional CNN-LSTM model with two parts: regional CNN and LSTM, in order to forecast the VA ratings of texts. The suggested regional CNN divides an input text into many areas, as opposed to a traditional CNN, which considers the entire text as input. This allows the important information of region to be extracted and weighted suitably and granted to the VA prediction. For VA prediction, such data is sequentially combined across regions.regional data contained inside sentence and long-distance dependencies between phrases taken into account in the forcast process by combining the regional CNN and LSTM.

A. Regional CNN-LSTM Model

1. Convolutional Layer :

Each regions local n gram features are initially extracted using a convolutional layer. In a region matrix $M \in \mathbb{R}^{d\times|V|}$ where $|V|$ is the vocabulary size of a region and d is the dimensionality of the word vectors, all word embedding are stacked. For instance in Figure 4, the region matrices x^{ri}, x^{rj} and x^{rk} are created by combining the word vectors in the regions $r_i = \{w_1^{ri}, w_2^{ri}, \ldots, w_I^{ri}\}$,$\mathrm{r}_j = \{w_1^{rj}, w_2^{rj}, \ldots, w_I^{rj}\}$ and $r_k = \{w_1^{rk}, w_2^{rk}, \ldots, w_I^{rk}\}$ We employ local n-gram features that are learned using L convolutional kernels in each region. We employ local n-gram features that are learned using L convolutional kernels in each region. A kernel $F_l(1 \le l \le L)$ creates the feature map y_n^l in a window of ω words $x_{n:n+\omega-1}$ as follows

$$y_n^l = f(W^l \cdot x_{n:n+\omega 1} + b^l)$$

where b^l stands for the weight matrix and bias related to the kernel, and is a convolutional operator. F^l, is the kernel length, d is the word vectors dimension, and f is the ReLU function. When a kernel moves gradually from $x_{1:1}$ to $x_{N+1:N}$ we obtain the final feature maps $y^l = \{y_1^l, y_2^l, \ldots, y_{N+1}^l\}$ of kernel F^l. Given that the regions texts vary in length y^l might have different dimensions for various texts. As a result, we refer to the dimension N as the maximum length of the CNN input in regions. In the event that the input length is less than N, zero vectors will be added. Each convolutional layer generates feature maps by passing the input vector to different kernels of various colors

$$Y = \{y^1, y^2, \ldots, y^L\} \in \mathbb{R}^{(N+\omega-1)\times L}$$

2. Max Pooling Layer

Max pooling samples the convolutional layer's output. Applying a max operation with a pooling size s to the output of each kernel is the most typical method for performing pooling. To preserve the most important information, the local dependency of various regions can get through max-pooling layer. After being flattened to a vector, the obtained region matrix is fed to the sequential layer.

3. Sequential Layer

The sequential layer sequentially integrates each vector of a region into a text vector in order to capture long-distance dependencies between regions. for vector composition

of sequential layer, LSTM is introduced. The final hidden state of the sequential layer which the LSTM cell passes through all regions is taken into consideration for VA forcasting text representation is taken consideration.

B. Region Division Strategy

1. Sequential Division Strategy

Using sequential technique, make every individual sentence in the text as an area, is one straightforward method. For example, if a text comprises 3 sentence, all three zones will receive those sentences. The most crucial information is extracted from each region using a separate convolutional and max-pooling layer before being fed into a global sequential layer with three LSTM recurrent units. This tactic is simple to use, but with a large sentence length margin, it will be quite unbalanced. The important characteristics in huge sentence going to be difficult to get .

2. Tree-Structured Division Strategy

A alternate method that more accurately captures the meaning of the text than the sequential method is to parse the input text as a tree-structured topology. A given text can be separated into regions based on the tree depth of the parse tree. These areas could be words, phrases, clauses, sentences, or even an entire paragraph—all linguistic expression function blocks.

F. Gate Recurrent unit

The 3 gate not take care of the internal state. The data that is kept in an LSTM recurrent unit's internal cell state is incorporated into the gated recurrent unit's hidden state. The group of data taken by next Recurrent unit. Types of GRU gates are

1) Update Gate(z): It decide what amount previous data to be transfered ahead. It is similar to an LSTM recurrent unit's Output Gate.
2) Reset Gate(r): Previous information to be discarded for this purpose this gate is used.Gated recurrent unit gates of forget and input have similar as of LSTM gates.
3) Current Memory Gate(h_t): In gated recurrent unit network, it is not considered to be. The Input Modulation Gate is a part of the Input Gate and is used to provide some non-linearity in input and to make the input Zero-mean, it incorporated in the Reset Gate. Being the part of reset gate decreases the effects of information that to be send from previous to future.

IV. Experimental Analysis

A. Experimental setup

In Google Colab, all three architectures have been implemented, creating a speedier computing environment.The reduction happen during the training period of the architecture ceased reducing after 8 epochs, so we trained the CNN architecture model for 8 epochs with a batch size of 782. We trained an LSTM network with a batch size of 782 for 5 epochs for the same reason. LSTM-CNN network batch size is of six epochs. The Adam optimizer was used to reduce the loss function, which was determined by Binary Cross-Entropy . To prevent overfitting in the network, we applied the Dropout approach.

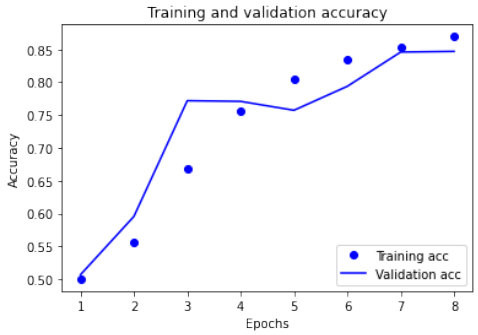

Fig. 2: Accuracy of LSTM

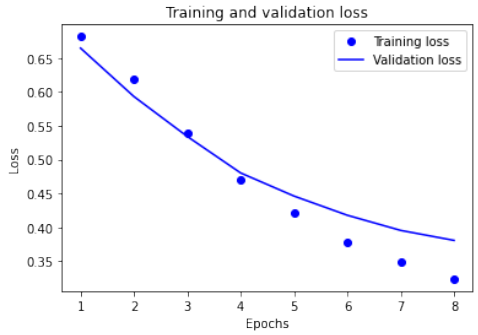

Fig. 3: Loss of LSTM

B. Results and Performance Analysis

The set of data addresses the issue of binary categorization because IMDb feedback can be either favourable or negative. IMDb movie reviews can be either favourable or unfavourable, hence the dataset addresses the binary categorization issue. Fig. 2 describes Accuracy of LSTM model. Lstm model accuracy is much better and less overfitting as compare to other model. In which during Training perform well but during validation accuracy were less.In 8 epochs Training and validation accuracy where not much difference could be seen. Fig. 3 descibes Loss of LSTM model whose training and validation loss is shown.

Fig. 4 describes the accuracy of CNN model.In which CNN during training performing well but during validation it not perform.In graph at every epoch training and validation accuracy where seen to be opposite of each other.On 8 epoch huge difference could be seen in graph between training and validation.

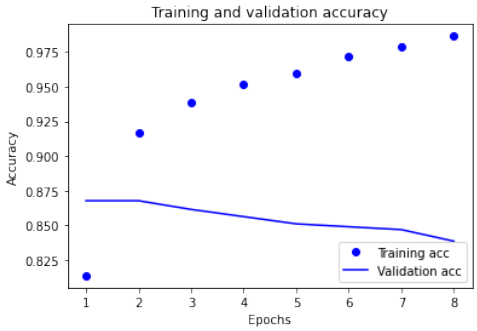

Fig. 4: CNN Accuracy

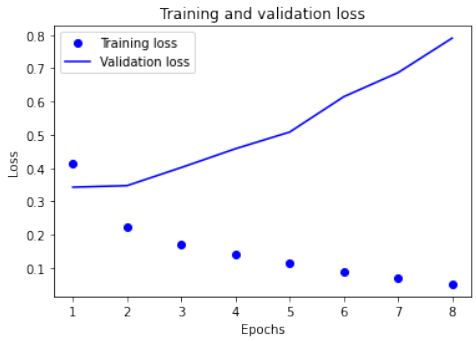

Fig. 5: CNN Loss

From Fig. 5 it could be seen Loss is between 2 to 8 epoch validation and training loss is increasing.

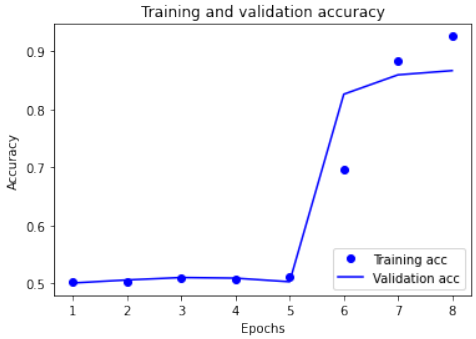

Fig. 6: Accuracy of CNN-LSTM

Metric	CNN	CNN-LSTM	GRU	LSTM
Accuracy	85%	85%	50%	87%
Loss	80%	39%	71%	34%

TABLE I: Comparision of deep learning Algorithm

Fig. 6 represents accuracy of CNN-LSTM model in which it could be notice that between 1 to 5 epoch accuracy where similar of training and validation but from 6 epoch there variation could be seen between training and validation epoch.

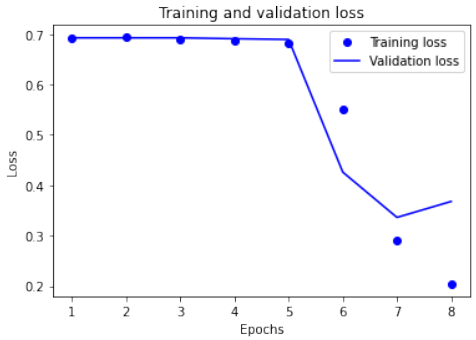

Fig. 7: Loss of CNN-LSTM

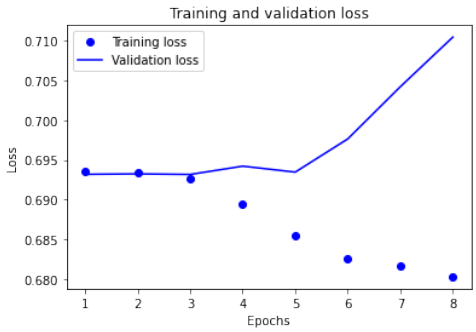

Fig. 8: Loss of GRU

Fig. 7 represents Loss of CNN-LSTM model. In which it could be observe that between 1 to 5 epoch loss is of similar of training and validation but after 6 epoch loss were decreasing but of validation but for training epoch loss still not much difference could be seen accuracy plot of gate recurring neural network is as shown in Fig. 9. Training accuracy is far better than validation accuracy here over-fitting occurs.Training and validation loss of GRU neural network is as shown in Fig. 8, here training loss is less as compare to validation loss

V. CONCLUSION

As the volume of online data grows rapidly sentiment analysis is becoming increasingly crucial.Sentiment analysis of internet reviews or social media is necessary for forecasting and predicting public opinion. From accuracy plots for all models given in section IV and TABLE I it is concluded that LSTM architectures accuracy is 2% more than CNN-LSTM and its loss is also 5% less than CNN-LSTM. Though CNN-LSTM architecture reduces a training time as compare to individual LSTM. LSTM perform very well compare to CNN, LSTM and GRU architecture. The model trained and tested for 8 epoch and adam optimizer is used for faster convergence and binarycross entropy is used as a loss function.

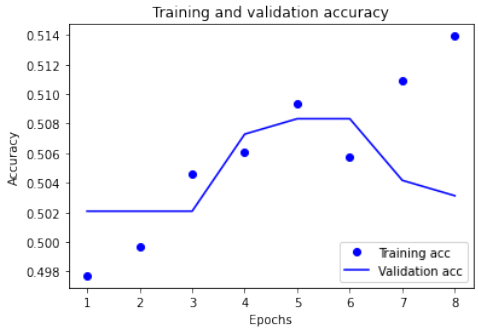

Fig. 9: Accuracy of GRU

VI. Acknowledgement

I would like to thank department head **Dr. Deepak Marathe** and thesis supervisor **Prof. Amol Patil** for providing their valuable technical suggestion on this project

References

[1] Rana S. and Singh A, "Comparative analysis of sentiment orientation using SVM and Naive Bayes techniques", *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)* . IEEE, 2016, pp. 106-111

[2] https://www.analyticsvidhya.com/blog/2021/07/word2vec-for-word-embeddings-abeginners-guide

[3] Kim Y, "Convolutional neural networks for sentence classification" *arXiv preprint arXiv*, 1408.5882, 2014

[4] Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions.", *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 6.02*, 1998, pp. 107-116.

[5] Li D. and Qian J, "Text sentiment analysis based on long short-term memory" *2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI)*, IEEE,2016

[6] https://towardsdatascience.com/introduction-to-deep-learning-with-keras-17c09e4f0eb2

[7] Maas A. L., Daly R. E., Pham P. T., Huang D., Ng A. Y. and Potts C., "Learning word vectors for sentiment analysis." *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, volume-1 Association for Computational Linguistics, 2011, pp. 142-150.

[8] Pennington J., Socher R. and Manning C., "Glove: Global vectors for word representation." *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532-1543.

[9] Xu B., Wang N., Chen T. and Li M., "Empirical evaluation of rectified activations in convolutional network." *arXiv preprint arXiv:1505.00853*, 2015

[10] W. L. Wei, C.H. Wu and J.C. Lin, "A regression approach to affective rating of Chinese words from ANEW", *Proc. Int. Conf. Affect. Comput. Intell. Interact.*, pp. 121-131, 2011.

[11] N. Malandrakis, A. Potamianos, E. Iosif and S. Narayanan, "Kernel models for affective lexicon creation", *Proc. Annu. Conf. Int. Speech Commun. Assoc. Interspeech*, pp. 2977-2980, 2011.

[12] L. C. Yu, J. Wang, K. R. Lai and X. Zhang, "Predicting valence-arousal ratings of words using a weighted graph method", —it Proc. 53rd Annu. Meeting Assoc. Comput. Linguist., pp. 788-793, 2015.

[13] J. Wang, L. C. Yu, K. R. Lai and X. Zhang, "Community-based weighted graph model for valence-arousal prediction of affective words", *IEEE/ACM Trans. Audio Speech Language Process*, vol. 24, no. 11, pp. 1957-1968, Nov. 2016.

[14] G. Paltoglou and M. Thelwall, "Seeing stars of valence and arousal in blog posts", *IEEE Trans. Affect. Comput.*, vol. 4, no. 1, pp. 116-123, Jan.–Mar. 2013.