EASR

Engineering and Applied Science Research

https://www.tci-thaijo.org/index.php/easr/index

Published by the Faculty of Engineering, Khon Kaen University, Thailand

# A hybrid model using MaLSTM based on recurrent neural networks with support vector machines for sentiment analysis

Srinidhi H*[1)], Siddesh GM[1)] and Srinivasa KG[2)]

[1)]Department of ISE, Ramaiah Institute of Technology (MSRIT), Bangalore-560054, India, Affiliated to Visvesvaraya Technological University, Belagavi, Karnataka, India
[2)]National Institute of Technical Teachers Training & Research, Chandigarh, India

Received 31 October 2019
Revised 13 January 2020
Accepted 24 January 2020

**Abstract**

Sentiment analysis is an ongoing research area in the field of data science. It helps in gathering insights into the behaviors of the users and the products associated with them. Most sentiment analysis applications focus on tweets from twitter using hashtags. However, if the reviews are taken by themselves, more clarity on the sentiments behind them is available. The primary challenge in sentiment analysis is identifying keywords to determine the polarity of the sentence. In this paper, a hybrid model is proposed using a Manhattan LSTM (MaLSTM) based on a recurrent neural network (RNN), i.e., long-short term memory (LSTM) combined with support vector machines (SVM) for sentiment classification. The proposed method focuses on learning the hidden representation from the LSTM and then determine the sentiments using SVM. The classification of the sentiments is carried out on the IMDB movie review dataset using a SVM approach based on the learned representations of the LSTM. The results of the proposed model outperform existing models that are based on hashtags.

**Keywords:** Recurrent Neutral Networks (RNN), Long Short-Term Memory (LSTM), Sentiment analysis, Support Vector Machines (SVM), MaLSTM

## 1. Introduction

Human interpretation of sentiment is definitely the most mature and accurate judge of sentiments. However, the big question in data science is how to embed this intelligence into machine learning models. The model behind sentiment analysis should be independent of the variations in semantics of the words used in the sentences that express the same idea. Recently, interest has turned towards learning a unique fixed-length representation of words from a large corpus of text - word embedding. Recurrent neutral networks (RNN), especially the long short-term memory (LSTM) network, which is better suited for long-term dependencies, has been particularly successful in using word embedding for complex tasks such as language translations [1] and text classification [2]. Given an input sequence (x1, ..., xT ), the LSTM sequentially updates a hidden-state representation using a memory cell containing four component vectors, a memory state, an input gate that controls information that gets stored in the memory cell, a forget gate [3-4] which determines the information to be omitted from a memory cell and an output gate that determines the manner in which the memory state affects other units.

RNNs are Turing complete [5] that is, it can be shown that for any commutable function, there exists a finite RNN capable of computing it. While, theoretically, RNNs are powerful learning models, in practice it is hard to train them owing to the vanishing gradient and exploding gradient problems [5]. LSTMs on the other hand, are capable of learning long-term dependencies. The key to this unique ability of an LSTM network is through the use of memory cell units. Similar to RNNs, a LSTM learns by sequentially updating its hidden-state representation. However, these updates also depend on a memory cell constituted by four real-valued vectors. At each time-step $t \in \{1, ..., T\}$ a LSTM performs updates which are parameterized by a number of weight matrices including Wi, W$f$ , Wc, Wo, Ui, U$f$, Uc, Uo and bias-vectors, bi, b$f$ , bc, and bo.

In this paper, the proposed system uses a support vector machine (SVM) with a radial basis function (RBF) kernel for the textual classification of positive and negative sentiments. The SVM takes the input vector and maps it to a higher dimension feature space to find the margin that is optimal to separate the two classes of sentiments. The mapping of features to higher dimensions is carried out using a kernel trick, as shown in Equation 1, where l is the number of support vectors, b is a bias term, $y_n \in \{-1, 1\}$ is the class sign to which the support vector belongs and $\alpha$ is obtained as the solution to the following quadratic problem, as shown in the Equation 2. In Equation 2, W represents the feature vector considered, $\xi$ represents a slack variable for the

*Corresponding author.
Email address: srinidhi.hiriyannaiah@gmail.com
doi: 10.14456/easr.2020.26

misclassification rate and to show that the decision boundary $y_i$ using a function of Φ that will not cross 1- ξ.

$$f(x) = \text{sign}\left(\sum_{n=1}^{l} y_n\, a_n\, .\, k(n,n) + b\right) \quad (1)$$

$$\begin{aligned} &min\frac{1}{2}W^T.W + C.\sum_{n=1}^{l}\xi_i \\ &s.t\; y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i \\ &\xi_i \geq 0, i = 1,..p \end{aligned} \quad (2)$$

Ideally, the number of support vectors should be a relatively small portion of the dataset. The equation of the RBF kernel is as shown in the Equation 3, where $\|x - x'\|^2$ is the squared Euclidean distance between the two feature vectors x and x' [6]. σ is the parameter that shows how far the influence of the training reaches.

$$K(x, x') = exp(\frac{\|x-x'\|^2}{2\sigma^2}) \quad (3)$$

The main aim of this paper is to show that combining LSTM with SVM on trained labeled data can be used to learn highly structured sentence representations for capturing the sentiments expressed by these sentences. Despite its simplicity, the proposed method significantly surpasses the current state-of-the-art methods for sentiment evaluation. Formally, we consider a supervised learning set where each training examples consists of a sequence (x1(a), ..., xT (a)), of fixed-size vectors along with a single label y. Sequences present in the dataset may be of different lengths. The motivating example considered is the task of classifying sentences, as 'positive' or 'negative' sentiments labeled by humans as y. In this case, each xi(a) denotes the vector representation of a word i from the sentence. By pre-training with a Siamese LSTM having the goal to learn semantic similarity between sentences, we then add a SVM classifier over the learned representations of the LSTM with the explicit goal to capture the sentiment of the given sentence(s). Multiple sentences in the paragraphs forming a mean pooling layer are added and then passed to the SVM for classification.

The rest of the paper is organized as follows. In Section 2, related work on the sentiment analysis with different existing models is discussed. It is followed by the methodology of the proposed hybrid MaLSTM model on sentiment analysis in Section 3. In Section 4, evaluation methodology of the sentiments is discussed. The training methodology of the MaLSTM model is discussed in Section 5. The experiments and results of the proposed MaLSTM model are discussed in Section 6, followed by the conclusions and future work.

## 2. Related work

Deep learning methods are used in the fields of computer vision, natural language processing and for image reconstruction and processing applications. However, they are also used for sentiment analysis. In this section, some of the related work on the sentiment analysis using deep learning and traditional machine learning methods are discussed.

Combining a recurrent neural network (RNN) with support vector machines (SVM) for time series classification [7] was applied to automatic Arabic speech recognition. The method used was to combine Echo State Networks [8] with support vector machines. This approach helped utilize a large reservoir by mapping input vectors at different time steps to the output of the reservoir, collecting them in a vector, and finally passing them to the SVM for classification. The method proved to be particularly useful for multiple label classification tasks. In [9] the authors proposed a recurrent SVM model, called Evoke, for the classification of the context sensitive language.

In [10], the idea of recurrent support vector machines is proposed that outperform gradient based recurrent neural networks on various time series tasks. They state that the gradient based information for backpropagation while training RNN is not enough. This is due to numerous local minima. The proposed model in [10], Evolino, achieves better accuracy than echo state networks. In [11] the authors proposed a variant called an intrinsic recurrent support vector machines which use internal memory to represent the current state of the system, just like RNNs. This is implemented by adding another weight vector to the present standard non-recurrent SVM. The model was useful at tasks such as summation and superimposed sine prediction.

In [12], the authors try to embed sentences using LSTM. The LSTM was trained on user-click through data logged by a commercial web search engine. This model learnt to attenuate the stop-words without explicitly specifying them. It was applied for information retrieval, where it outperformed several existing state-of-the-art methods. In [13], a semi-supervised sequence learning model is proposed by first trying to predict what comes next in a sequence, and a second algorithm used a sequence auto-encoder. These two algorithms were used as a "Pre-training" step for a later supervised sequence-learning algorithm.

Tree LSTMs are tree structured network topologies that are used for generalization of standard LSTM. In tree LSTMs, each sentence is first transformed into its corresponding parsed tree. It computes the hidden state at a given nodes using the states of all the child nodes. The advantage of using tree LSTMs is that a tree structured network propagates the relevant information better than a sequential LSTM. It better helps in finding the similarity of the sentences compared to the model proposed in [14], except that the input sentence representation are now generated by Tree-LSTMs rather than skip-thoughts.

There has been extensive research in developing hybrid models for sentiment analysis. In [15], the authors proposed a model using a deep neural network (DNN) and SVM for classification. The distinguishing factor of this model is that SVM is used for classification rather than the multinomial logistic regression with a softmax function at the top layer. Here, the posterior probabilities are replaced by the SVMs capability of drawing hyperplanes for classification rather than using hidden Markov models. The model selects the representation learning objective to directly reflect the given semantic similarity labels, which we use for pre-training. While all the aforementioned neural network approaches utilize complex learners to predict semantic similarity from the sentence representations, we impose much stronger demands. The learned representation space is only truly semantically structured if a simple metric suffices to predict sentence similarity, which is subsequently used for sentiment evaluation.

An empirical comparison of SVM and artificial neural networks (ANNs) for sentiment analysis are discussed in [16]. The results demonstrated that the ANN performed better than the SVM. In [17], the authors presented a cache based LSTM model that captures the semantic information in text. The memory in the LSTM model is divided into groups to capture global semantic features using different forgetting rates. A simple weighted sum model for capturing context sensitive information for sentiment analysis is

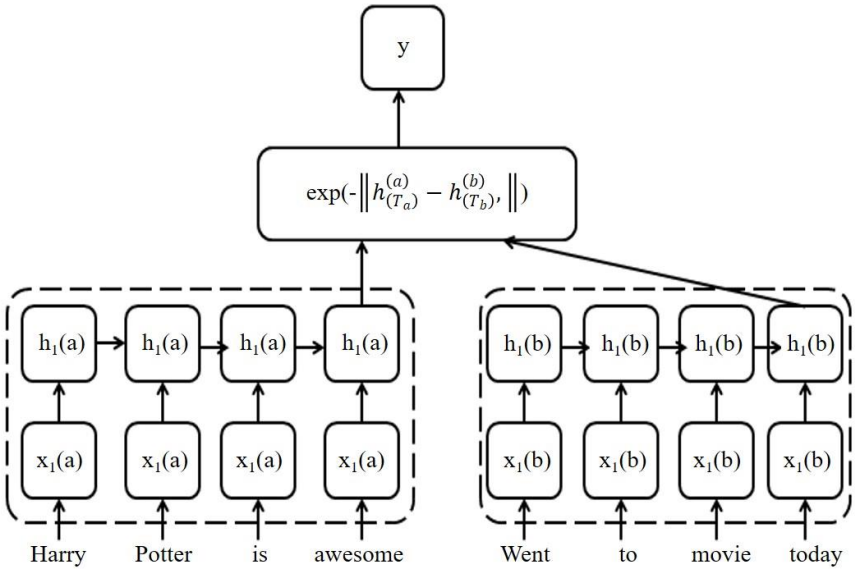


**Figure 1** Siamese architecture (pre-training) for feature extraction of sentences in a single-vector form

proposed in [18-19]. This helps in learning the negation of lexicon sentiments in the sentence. Aspect based sentiment analysis has been proposed in [20] as it has many applications in natural language processing [21].

The Siamese recurrent neural network [22-23] uses distinct hidden units to encode specific characteristics of a sentence. Then, the trained MaLSTM is able to infer semantic similarity between sentences by simply combining their differences in various characteristics [24-25]. The Siamese recurrent neural network involves simultaneous training of two recurrent neural networks with symmetric weights. Like these methods, the proposed model represents sentences using neural networks whose inputs are word vectors learned separately from a large corpus [25]. The proposed model is an extension of the Siamese recurrent neural network that is used to evaluate semantic similarity between sentences.

Sentiment analysis research is an evolving area in the field of computer science [26]. A case study of sentiment analysis of airline tweets is discussed in [26], where customer satisfaction levels are analyzed using the tweets of the airline data. Deep learning methods are used in sentiment analysis such as RNN with bag-of-words and word embedding techniques [27]. The exploration of the vanishing gradient problem in neural networks for sentiment analysis is a great challenge and needs to be carefully addressed. To eliminate the vanishing gradient in $l_2$-based models, the idea is simply to advocate that each phrase encodes a "thought" which we can equivalently encode as a vector. However, each word also encodes a thought, and thus each network learns to combine the thoughts in a group into a single underlying thought (which is the output of the last hidden unit of the LSTM). In this paper, the proposed MaLSTM model hidden layers is visualized to extract the prominent features for sentiment analysis.

## 3. Proposed hybrid model using MaLSTM for sentiment analysis

The proposed hybrid model using the MaLSTM model employs a combination of RNN and SVM. The focus is on weights of the Siamese architecture with tied weights such that $LSTM_a = LSTM_b$, whose output is a 50 length vector, with the input of each word a 300 length vector, from the Gensim word2vec model [28]. A sentence with T words is represented by the LSTM using $h_T$, the final hidden state vector after the LSTM has performed the updates for t = 1, ..., T. In the MaLSTM model, the similarity of the sentences is calculated as shown in Equation 4. It represents a function g() that calculates the similarity between the hidden layer units $h_{(Ta)}$ and $h_{(Tb)}$ of the MaLSTM model, as shown in Figure 1. The $\epsilon$ class is either 0 or 1 based on the similarity calculated using the exponentiation function.

$$g\left(h_{(T_a)}, h_{(T_b)}\right) = \exp(-\left\|h_{(T_a)}^{(a)} - h_{(T_b)}^{(b)}, \right\|)\varepsilon(0,1) \quad (4)$$

The important advantage of the combination of RNN+SVM in the MaLSTM model is that it behaves like an encoder, unlike the typical RNNs used in language models, which aim to predict the next word given a sequence of the previous words. The error signal that is propagated in the model depends only on the function g(). It minimizes the problem of the vanishing gradient as in $l_2$-based models erroneously believe that semantically different sentences are almost identical during the early stages of training.

A simple Manhattan distance is used in g which compels the LSTM to completely capture the semantic differences during training, rather than supplementing the recurrent network with a far more complex learner that can subsequently resolve some of the flaws in the learned representations. The proposed model is similar to the Siamese architecture for face verification as proposed by [29] as shown in Figure 1, but it employs symmetric LSTMs rather than ConvNets. Siamese neural networks have been proposed for a variety of language-related metric learning tasks [30], but to our best knowledge, recurrent neural networks remain largely unexplored in the current context. For training, we started with the same weights for both $LSTM_a$ and $LSTM_b$. Training is done in batches, so the output of each LSTM is in the form (Batch − Size, 50). The negative exponent of the absolute value of the row-wise differences across the output of either LSTM is computed. The obtained value is then compared with the actual label y, and the mean squared error (cost function) is calculated. We then take the gradient with respect to the weights of either of the LSTMs. Then, backpropagation is done with the

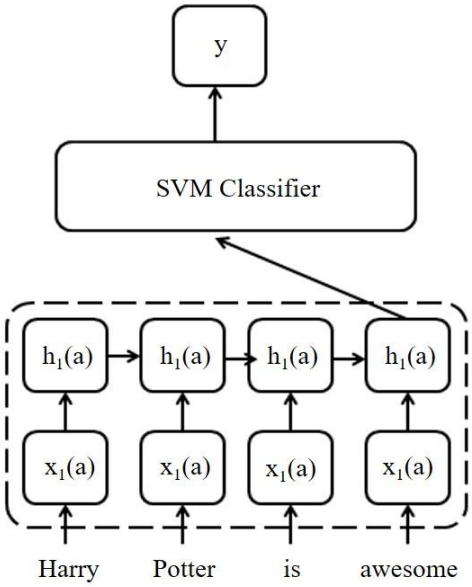


**Figure 2** MaLSTM model with LSTM feature extraction of sentences and SVM classifier

optimizer [31], and the gradient for backpropagation is computed as shown in Equation 5, where the grad (LSTMa) and grad (LSTMb) represents the gradients calculated by each unit in the MaLSTM model.

This helps symmetric backpropagation of the LSTMs thus assists in maintaining the criteria for $LSTM_a$=$LSTM_b$. After extracting the feature vector by pre-training the Siamese LSTM on the SemEval 2014 [32] dataset, we take the output of the last hidden unit of the LSTM, which is a single vector. The learned representations of a sentence are stored in the feature vector [33]. Here, the weights of both LSTMs represented in Figure 1 are tied, i.e., $LSTM_a$ == $LSTM_b$. In this case, there are cost functions used other than $\exp(-\|h(a) - h(b)\|)$, such as sigmoid or cosine similarity. We need to examine the diagonal of the dot product of the outputs of $LSTM_a$ == $LSTM_b$.

$$Gradients = \frac{Gradient\ (LSTM_a) + Gradient\ (LSTM_b)}{2} \quad (5)$$

The sentences are converted to their embedding [34-35] matrices (300 dimension vectors), and the output of the LSTM is a single 50 length vector. We then add a SVM classifier to the output of the last hidden unit of the LSTM as a binary classification task. This classifier can also be used for multiple classification tasks as well.

We average of the vector representations of each sentence, i.e., the output of the last hidden unit of the LSTM for each sentence to evaluate our model on the IMDB movie review dataset [36]. This has a paragraph with multiple sentences and a single label. Then, a mean pooling layer was applied, thus, giving a single feature vector as shown in Figure 2. Figure 3 represents the LSTM + Mean Pooling + SVM classifier for sentiment analysis of the IMDB movie review dataset [36].

In summary, the entire process of the MaLSTM model is as follows:

1. Pre-training: Train the SemEval dataset with the Siamese LSTM architecture as shown a y in Figure 1. We ensure that the weighs of both LSTMa and LSTMb are the same by averaging out both their gradients during backpropagation.
2. Feature extraction: Extract the feature vectors of sentences of the Kaggle challenge using LSTMa. The feature vector of a sentence is the output of the last hidden unit of the LSTM.
3. Classifier: Using the extracted features (hidden representations of LSTMa) as training data, train the model with a SVM classifier for sentiment evaluation.

## 4. Sentiment evaluation using the MaLSTM model

The dataset of [36] considered for the evaluation contains 40,138 sentence pairs with a 7,086/33,052 training/test splits. Each pair in the dataset is annotated with a label that corresponds to either a positive or negative sentiment. A RNN, despite training on a large corpus for two weeks, was unable to distinguish the set of test sentences. This highlights the difficulty of each task. The RNN model was not able to distinguish even if each of the sentences has been labeled as one of the two classes, i.e., positive or negative, for predicting the set of test sentences.

For the purpose of this task, the learned representation of the Siamese LSTM network was used to perform the classification. Specifically, the learned representation of a sentence is computed from the MaLSTM representations, h(a), of a given sentence, which is a feature vector. Over these features, we train a SVM, which uses a radial-basis-kernel, to classify entailment (with hyperparameters optimized using 5-fold cross-validation). The proposed MaLSTM model takes input word-vectors that have been learned from an external corpus. A 300-dimension word2vec embedding is used that is able to capture complex inter-word relationships, such as vec(king) − §‘fi´vec(man) + vec(roman) = vec(queen) [30]. The proposed model does not make use of extensive manual feature generation beyond the separately learned word2vec vectors, unlike the SemEval 2014 results.

For the binary classification task of positive or negative sentiment, the learned representations for semantic similarity are used (fixed with no additional fine-tuning) to perform the classification. Specifically, we compute the following simple features from the MaLSTM representations h(a), of a given sentence. A radial-basis-kernel SVM is trained with hyper-

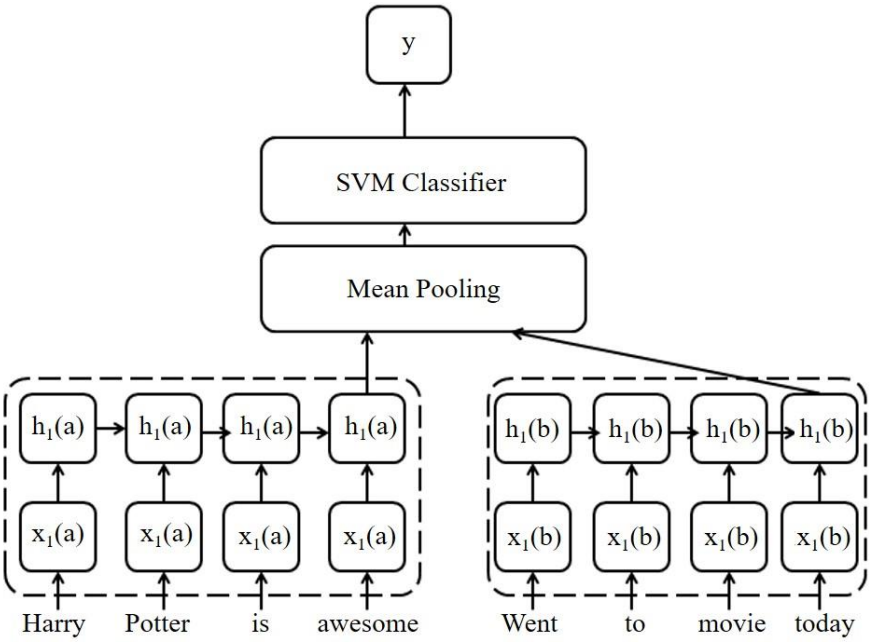


**Figure 3** MaLSTM model with a SVM classifier and mean pooling

**Table 1** Analysis of MaLSTM model on different metrics for Kaggle dataset

| Models | Precision | Recall | F-measure | Accuracy |
|---|---|---|---|---|
| Bayesian Network classifier (William Wilcox) | 91.83 % | 91.76 % | 88.94 % | 93.8% |
| Random Forest (Breakfast Pants) | 92.72 % | 91.46 % | 90.94 % | 93.26% |
| RNN (Rodger Devine) | 89.83 % | 91.76 % | 89.73 % | 92.8% |
| SVM (Danny Wu) | 88.83 % | 90.69 % | 89.64 % | 92.7% |
| **MaLSTM** | 94.63 % | 93.37 % | 92.47 % | **95.3%** |

parameters optimized using 5-fold cross-validation to classify sentiments on these features. Even though the provided features are trained for the distinct goal of semantic similarity scoring (with no information regarding sentiments), they capture important characteristics of the sentences that are highly useful for assessing sentiments.

Although the model was originally trained to understand semantic similarity between sentences (pre-training), the learned representations or the output of the last hidden unit, which is a single vector, has captured enough characteristics to carry out categorical classification of sentences.

## 5. Training a hybrid model using MaLSTM

The proposed hybrid model using a MaLSTM network employs 50-dimensional hidden representations, $h_t$, and memory cells, $c_t$. The optimizer used is from [37]. To prevent over-fitting, early-stopping is used based on a validation set containing 30% of the training samples. It is well-established that the performance of LSTMs depends largely on their initialization. The LSTM weights are initialized with Xavier initialisation. The forget gate bias is initialized with a separate value of 1.5 [38] for solving long range dependencies. Then, the proposed MaLSTM is pre-trained for the SemEval 2014 semantic textual similarity tasks. In the case that the bias of the forget gates $f$t is set to 0, the performance of the LSTM [39] on the cross validation dataset is worse. The resulting weights from this pre-training phase forms the starting point for the sentiment analysis data.

The hypothesis in [40] was used for the proposed MaLSTM to infer the semantic similarity between the sentences by aggregating their differences. Hence, we add a SVM classifier to the output of the last hidden unit of the LSTM. For the Kaggle dataset [41], a radial-basis-kernel SVM is used with gamma=0.8, and c=100. As for the IMDB dataset [41], a mean pooling layer is added that averages all of the sentence representations containing a single label, with gamma=2.5, c=100.

The entire training of the Siamese LSTM with end to end backpropagation was accelerated using the Nvidia GeForce GPU. The code was written in Python using the library, Theano, a GPU accelerated framework for machine learning in Python. The speed gained in training time on the GPU (along with Nvidia's cuDNN) ~~28~~was 15x faster than the CPU.

## 6. Experiments and results

The proposed hybrid model was evaluated on the Kaggle [41] and the IMDB movie review datasets. The result of the MaLSTM model applied on the Kaggle dataset [41] is shown in Table 1. The results of the proposed model are compared against the existing models, namely a Bayesian network classifier, random forest, RNN and SVM. Let X, Y, Z denote the positive, negative and neutral classes of sentiments identified. Then, the accuracy of the model is calculated using Equation 6 where, tpX, tpY and tpZ are true positives that are correctly classified. Other values, eXY, eXZ, eYX, eYZ, eZX and eZY are false positives that are incorrectly classified. The proposed system, MaLSTM, was measured with the other indices of precision, recall and F-measure, as shown in the Table 1. The names in parentheses refer to team names.

**Table 2** Class of reviews in the IMDB dataset

| Genre | Positive | Neutral | Negative |
|---|---|---|---|
| Thriller | 57% | 16% | 29% |
| Comedy | 73% | 8% | 19% |
| Action | 69% | 24% | 7% |
| Drama | 58% | 27% | 15% |

**Table 3** MaLSTM + mean pooling model score of the IMDB dataset

| Model | Precision | Recall | F-measure | Accuracy |
|---|---|---|---|---|
| Bayesian Network classifier (William Wilcox) | 79.86% | 78.36% | 76.53% | 82.9% |
| Random Forest (Breakfast Pants) | 67.65% | 66.72% | 66.63% | 68.4% |
| RNN (Rodger Devine) | 55.68% | 53.73% | 54.35% | 56.4% |
| SVM (Danny Wu) | 67.73% | 66.48% | 67.32% | 68.7% |
| MaLSTM + mean pooling | **94.83%** | **93.73%** | **93.72%** | **95.8%** |

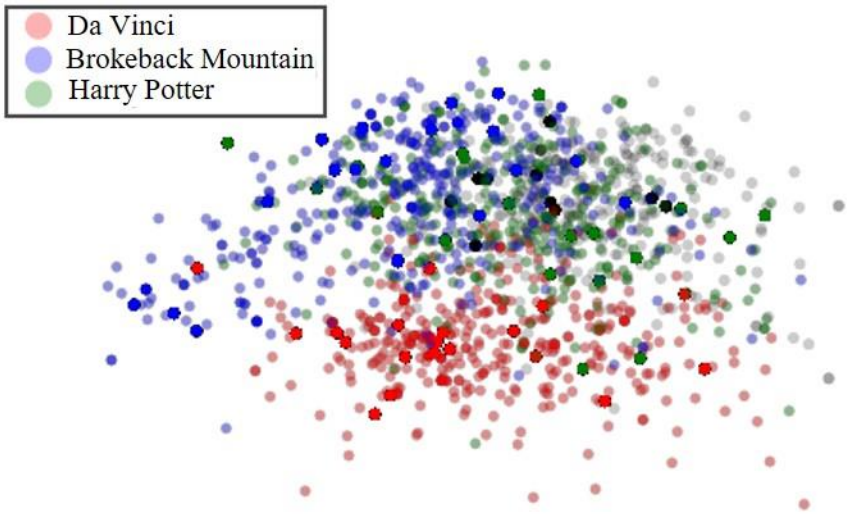


**Figure 4** PCA plot hidden representations of the Kaggle dataset reduced to two-dimensions

$$Accuracy = \frac{tpX + tpY + tpZ}{tpX + eXY + eXZ + tpY + tYX + tYZ + tpZ + tZX + tZY} \quad (6)$$

It can be observed from the Table 1 that the proposed system achieves higher accuracy compared to the existing models. The precision measure is also high for the MaLSTM model as the combination of RNN+SVM helps in the polarization of the sentiments. However, it should be noted that there is a slight increase in the recall compared to the SVM model. This is because the classification of the polarity is done using the SVM model. However, it increases the accuracy of the model.

In the case of the IMDB movie review dataset, as illustrated in Figure 3, a mean pooling layer is added over the output of the RNN for each sentence, as most samples contain multiple sentences. The percentage of positive, negative and neutral reviews based on the different categories are shown in the Table 2. It shows a sample of categories and the corresponding class of the reviews. For example, the genre 'comedy' has 73% positive reviews, 8% neutral reviews and 19% negative reviews.

The results of the MaLSTM model using mean pooling with the IMDB review dataset are shown in Table 3. The accuracy of the model is computed using Equation 6. It can be observed from the table that the proposed MaLSTM model combined with mean pooling gives better accuracy compared to the other models. The model shows an increased accuracy of 0.5% compared to the Kaggle dataset. It outperforms the existing models as well.

### *6.1 Sentence representations of MaLSTM*

It is important to see how the MaLSTM model performs in learning the representation of sentences and classifies their polarity. The hidden unit representation of the proposed MaLSTM is investigated in this section, which is a vector of length 50 representing a sentence. Visualizing hidden units requires non-linear dimensionality reduction methods like t-SNE [42]. Principal component analysis (PCA) is used for visualizing the hidden unit representations of the LSTM in a 2D plane.

Figure 4 shows a PCA plot of the Kaggle dataset. In general, most of the sentences had 3 themes, namely, Da Vinci, Harry Potter and Brokeback Mountain, around which all the sentences were based yielding positive or negative sentiments. As can be seen from the plot, there are three distinct clusters. The model is able to classify the sentences from the vector information available from the Siamese LSTM, and is independent of the fact that the three highlighted themes could be books or even movies.

Figure 5 is a PCA plot of the IMDB movie review dataset, which has 25,000 samples. This plot was colorized using K-means. The representations towards the left ($x < -0.2$) had mixed reviews. The plot towards the right

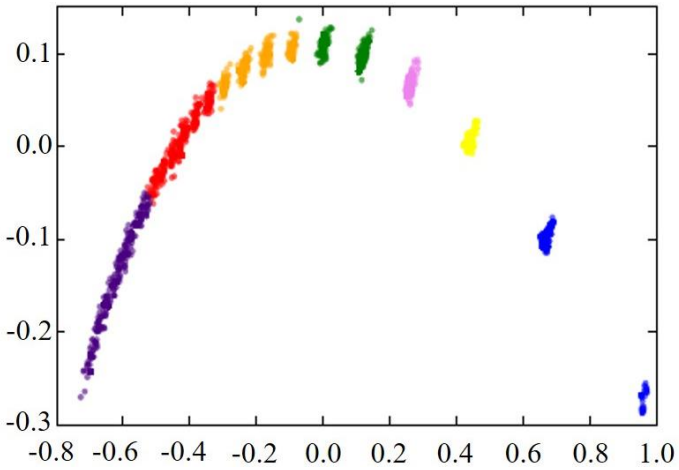


**Figure 5** PCA plot of IMDB movie review dataset

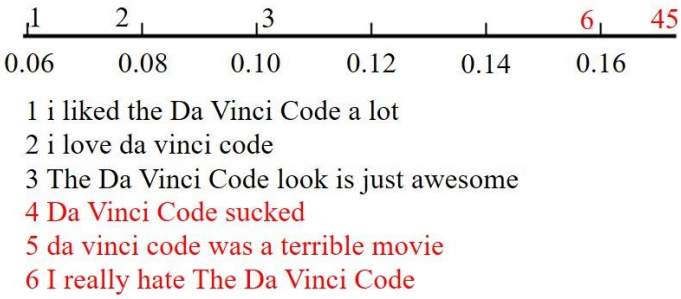


**Figure 6** 17th hidden unit of a sentence representation based on reviews of 'Da Vinci' in MaLSTM model

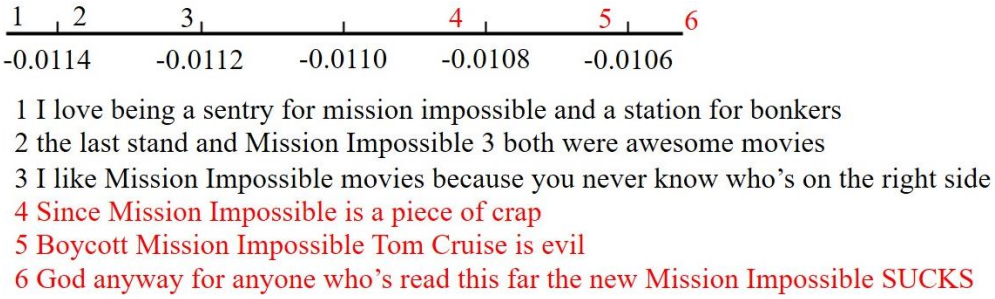


**Figure 7** 17th hidden unit of a sentence representation based on reviews of 'Mission Impossible' in MaLSTM model

(x > 0.2) have more distinct reviews (easier to classify). All the highlighted themes in Figure 4 are just the prevailing ones. For example, Mission Impossible samples were few. So, the sentences not belonging to the prevailing three themes are depicted in grey. Thus, the MaLSTM has learnt the sentence meaning in the form of a 50-length vector. It is able to apply sentiment analysis on these learned representations very well.

The hidden unit representations of the MaLSTM model are shown in Figures 6, 7 and 8. They represent the encoded nature of sentences using the MaLSTM model. It is clear that the 17th hidden unit shown in Figures 6 and 7 has learnt to detect sentiment of sentences. The sentences containing negative words are found towards the left in Figures 6 and 7. Various hidden units are particularly sensitive to the direct objects, separating sentences describing actions on balls, grass, cosmetics, vegetables, and even emotions.

The categorical nature of the sentences is also identified by the proposed MaLSTM model. The 48th hidden unit of the LSTM learns the categorical nature as shown in Figure 8. It represents categorical themes of a sentence, such as watching movie, reading book, or comments about a movie or book. The books tend to be towards the right, and the movies to the left. It can be seen that the model captures semantics perfectly and can be useful for multiple classification problems as well, other than sentiment.

## 7. Conclusions

Sentiment analysis plays an important role in the online world in e-commerce applications. Deep learning methods with RNN help in analyzing complex relationships among the different entities in the datasets considered for sentiment analysis. The proposed system (MaLSTM) demonstrates that an adaptation of the Siamese LSTM with SVM forms a good model for text classification. In cases with many sentences, a mean pooling layer is added and then passed to the SVM for classification. Since the model can find hidden unit

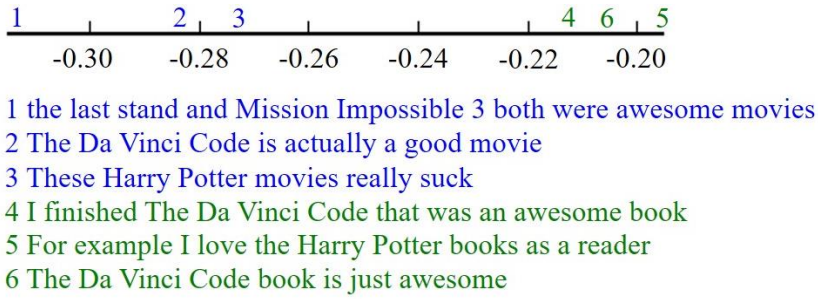


**Figure 8** 48th hidden unit of sentence representation using the MaLSTM model for category identification

representations of sentences, it can be deployed in real-time applications. The results on several text classification tasks show that the method is competitive with state-of-the-art methods.

In the future, the learned representations of a sentence gained using the MaLSTM model can form a good foundation for data analytics due to its simple and interpretable structure. Classification can be improved with regularization strategies in neural network models using dropout. However, in the future, real-time deployment of the model can be considered as well.

## 8. Acknowledgements

This research was supported by Ramaiah Institute of Technology (MSRIT), Bangalore, India and Visvesvaraya Technological University, Jnana Sangama, Belagavi, India.

## 9. References

[1] Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. Neural Information Processing Systems 27 (NIPS 2014); 2014 Dec 8-13; Montréal, Canada. p. 3104-12.

[2] Graves A. Supervised sequence labelling. In: Kacprzyk J, editor. Supervised sequence labelling with recurrent neural networks. Berlin, Heidelberg: Springer; 2012. p. 5-13.

[3] Gers FA, Schmidhuber J, Cummins F. Learning to forget: continual prediction with LSTM. Neural Comput. 2000;12(10):2451-71.

[4] Hochreiter S, Bengio Y, Frasconi P, Schmidhuber J. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In: Kolen JF, Kremer SC, editor. A Field Guide to Dynamical Recurrent Networks. USA: IEEE; 2001. p. 237-43.

[5] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Network. 1994;5(2):157-66.

[6] Tai KS, Socher R, Manning CD. Improved semantic representations from tree-structured long short-term memory networks. arXiv:1503.00075. 2015:1-11.

[7] Adya M, Collopy F. How effective are neural networks at forecasting and prediction? A review and evaluation. J Forecast. 1998;17(5-6):481-95.

[8] Lukoševičius M, Jaeger H. Reservoir computing approaches to recurrent neural network training. Comput Sci Rev. 2009;3(3):127-49.

[9] Schmidhuber J, Gagliolo M, Wierstra D, Gomez F. Evolino for recurrent support vector machines. arXiv:cs/0512062. 2005:1-10.

[10] Schmidhuber J, Wierstra D, Gagliolo M, Gomez F. Training recurrent networks by evolino. Neural Comput. 2007;19(3):757-79.

[11] Schneegaß D, Schaefer AM, Martinetz T. The Intrinsic Recurrent Support Vector Machine. ESANN'2007 proceedings - European Symposium on Artificial Neural Networks; 2007 Apr 25-27; Bruges, Belgium. p. 325-30.

[12] Palangi H, Deng L, Shen Y, Gao J, He X, Chen J, et al. Deep sentence embedding using long short-term memory networks: analysis and application to information retrieval. IEEE/ACM Trans Audio Speech Lang Process (TASLP). 2016;24(4):694-707.

[13] Dai AM, Le QV. Semi-supervised sequence learning. Neural Information Processing Systems 28 (NIPS 2015); 2015 Dec 7-15; Montréal, Canada. p. 3079-87.

[14] Kiros R, Zhu Y, Salakhutdinov RR, Zemel R, Urtasun R, Torralba A, et al. Skip-thought vectors. Neural Information Processing Systems 28 (NIPS 2015); 2015 Dec 7-15; Montréal, Canada. p. 3294-302.

[15] Zhang X, Zhao J, LeCun Y. Character-level convolutional networks for text classification. Neural Information Processing Systems 28 (NIPS 2015); 2015 Dec 7-15; Montréal, Canada. p. 649-57.

[16] Moraes R, Valiati JF, Neto WPG. Document-level sentiment classification: An empirical comparison between SVM and ANN. Expert Syst Appl. 2013;40(2):621-33.

[17] Xu J, Chen D, Qiu X, Huang X. Cached long short-term memory neural networks for document-level sentiment classification. arXiv:1610.04989. 2016:1-10.

[18] Teng Z, Vo DT, Zhang Y. Context-sensitive lexicon features for neural sentiment analysis. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing; 2016 Nov 1-5; Austin, Texas. USA: Association for Computational Linguistics; 2016. p. 1629-38.

[19] Zhang L, Wang S, Liu B. Deep learning for sentiment analysis: A survey. WIREs: Data Min Knowl Discov. 2018;8(4):e1253.

[20] Tang D, Zhang M. Deep learning in sentiment analysis. In: Deng L, Liu Y, editor. Deep Learning in Natural Language Processing. Singapore: Springer; 2018. p. 219-53.

[21] Al-Smadi M, Qawasmeh O, Al-Ayyoub M, Jararweh Y, Gupta B. Deep recurrent neural network vs support vector machine for aspect-based sentiment analysis of Arabic hotels' reviews. J Comput Sci. 2018;27:386-93.

[22] Nowak J, Taspinar A, Scherer R. LSTM recurrent neural networks for short text and sentiment classification. In: Rutkowski L, Korytkowski M, Scherer R, Tadeusiewicz R, Zadeh L, Zurada J, editors.

International Conference on Artificial Intelligence and Soft Computing; 2017 Jun 11-15; Zakopane, Poland. Springer; 2017. p. 553-62.
[23] Mueller J, Thyagarajan A. Siamese recurrent architectures for learning sentence similarity. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence; 2016 Feb 12-17; Arizona, USA. USA: AAAI; 2016. p. 2786-92.
[24] Mihalcea R, Corley C, Strapparava C. Corpus-based and knowledge-based measures of text semantic similarity. Proceedings of the 21st national conference on Artificial intelligence; 2006 Jul 16-20; Massachusetts, USA. USA: AAAI; 2006. p. 775-80.
[25] Rehurek R, Sojka P. Software framework for topic modelling with large corpora. Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks; 2010 May 22; Valletta, Malta. p. 46-50.
[26] Kumar S, Zymbler M. A machine learning approach to analyze customer satisfaction from airline tweets. J Big Data. 2019;6:1-16.
[27] Wazery YM, Mohammed HS, Houssein EH. Twitter sentiment analysis using deep neural network. 14th International Computer Engineering Conference (ICENCO); 2018 Dec 29-30; Cairo, Egypt. USA: IEEE; 2018. p. 1-6.
[28] Yih WT, Toutanova K, Platt JC, Meek C. Learning discriminative projections for text similarity measures. Proceedings of the fifteenth conference on computational natural language learning; 2011 Jun 23-24; Oregon, USA. USA: Association for Computational Linguistics; 2011. p. 247-56.
[29] Chen K, Salman A. Extracting speaker-specific information with a regularized siamese deep network. Advances in Neural Information Processing Systems 24 (NIPS 2011); 2011 Dec 12-17; Granada Spain. p. 298-306.
[30] Zeiler MD. ADADELTA: an adaptive learning rate method. arXiv:1212.5701. 2012:1-6.
[31] Li Y, Xu L, Tian F, Jiang L, Zhong X, Chen E. Word embedding revisited: a new representation learning and explicit matrix factorization perspective. Proceedings of the 24th International Conference on Artificial Intelligence; 2015 Jul 25-31; Buenos Aires, Argentina. p. 3650-6.
[32] Rosenthal S, Ritter A, Nakov P, Stoyanov V. SemEval-2014 Task 9: Sentiment analysis in Twitter. Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval '14; 2014 Aug 23-24; Dublin, Ireland. p. 73-80.
[33] Maas AL, Daly RE, Pham PT, Huang D, Ng AY, Potts C. Learning word vectors for sentiment analysis. Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies; 2011 Jun 19-24; Oregon, USA. USA: Association for Computational Linguistics; 2011. p. 142-50.
[34] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. Proceedings of the 13th International Conference on Artificial Intelligence and Statistics; 2010 May 13-15; Sardinia, Italy. p. 249-56.
[35] Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv:1603.04467. 2016:1-19.
[36] Kaggle [Internet]. 2015 [cited 2015 Jul 9]. Available from: https://www.kaggle.com/iarunava/imdb-movie-reviews-dataset.
[37] Chetlur S, Woolley C, Vandermersch P, Cohen J, Tran J, Catanzaro B, et al. cudnn: Efficient primitives for deep learning. arXiv:1410.0759. 2014:1-9.
[38] Maaten LVD, Hinton G. Visualizing data using t-SNE. J Mach Learn Res. 2008;9:2579-605.
[39] Hochreiter S, Schmidhuber J Long short-term memory. Neural Comput. 1997;9(8):1735-80.
[40] Pascanu R, Mikolov T, Bengio Y. On the difficulty of training recurrent neural networks. Proceedings of the 30th International Conference on Machine Learning; 2013 Jun 16-21; Atlanta, USA. p. 1310-8.
[41] Kaggle [Internet]. 2015 [cited 2015 Jul 9]. Available from: https://www.kaggle.com/c/si650winter11/data.
[42] Chopra S, Hadsell R, LeCun Y. Learning a similarity metric discriminatively, with application to face verification. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05); 2005 Jun 20-25; San Diego, USA. USA: IEEE; 2005. p. 539-46.