



Sense understanding of text conversation using temporal convolution neural network

Sandeep Rathor¹  · Sanket Agrawal²

Received: 22 March 2021 / Revised: 11 October 2021 / Accepted: 3 January 2022 /

Published online: 14 February 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

This paper proposes a model which uses Spatio Temporal features for real-time sense understanding of a text conversation. The proposed model uses CNN along with the concept of LSTM to create a new Spatio temporal cell. Furthermore, the proposed model is used to classify the sentences into eight senses. The model achieved an F-Score around 0.984 on sense classification. Additionally, the efficiency and capabilities of the model are also tested on a standard IMDB sentiment classification dataset. On the IMDB dataset, the model gave an accuracy of 89.27. The experimental results show that the proposed model works better than a CNN model, a Bi-LSTM model, and a combination of CNN & LSTM model in terms of a number of parameters and execution time.

Keywords Temporal CNN · Spatio temporal · Sense understanding · Text processing · Advance machine learning technique

1 Introduction

The basic entity for exchanging the information among the person is communication. It can be unidirectional or bidirectional. However, the most effective way of communication is bidirectional i.e. 2-way communication, generally referred to as a conversation. It is a dynamic and spontaneous activity, therefore, we cannot prepare our response before getting the other's response. The conversation can be in the form of a video, audio or text. However, the most effective and basic form of conversation is the text [5, 9]. As the

✉ Sandeep Rathor
sandeep.rathor@gla.ac.in

Sanket Agrawal
sanket.agrawal@infosys.com

¹ Department of CEA, GLA University, Mathura, India

² Infosys Ltd, Pune, India

present age is the age of information technology so, everyone shares their ideas or views through social media like Twitters, Facebook, E-mail, WhatsApp, etc. The ideas or views of individuals may reflect the various features like emotions, sentiments, etc. Despite these basic features, a conversation may reflect its sense about which it belongs like: social, medical, religious, political, educational, games & sports, advertisement, and legal. Therefore, in a real-time conversation between two or more persons, the proposed system can recognize the sense of communication about which it is taking place. Suppose, if people are discussing IPL cricket match through a WhatsApp conversation then the sense of this conversation is to be games & sports, while if two persons are discussing the treatment and prevention of coronavirus, then the sense of this conversation is to be medical. Therefore, based on the daily life conversation, a sense may be obtained into one of the various categories like social & personal, medical, religious, political, education, games & sports, advertisement, and legal. Sense understanding is one of the emerging fields of human-computer interaction. It is different from emotion and sentiment recognition by the number of points. The objective of emotion recognition is to recognize the emotions on the basis of conversation like happy, sad, joy, surprise, excited etc. while sentiment recognition deals with the analysis of sentiments in a conversation as positive, negative or neutral [22]. Nowadays sentiment analysis is required by various renowned companies to analyse the customer's choice and feedback about their product [3]. It may help to improve the quality of the product as per the customer's choice or requirements.

The focal dispassionate of the proposed research is to recognize the sense of a real-life text conversation. It can be utilized in various real-life applications of human-computer interaction. For example, suppose two persons are planning to impose a fraud case on a specific person then the sense of this conversation will be under the category of "Legal" and it can be recognized by our proposed system. The temporal convolution technique is best suited to process the text or natural language efficiently [11, 19, 21]. Therefore, temporal CNN is used in the proposed model to recognize the sense of the conversation efficiently.

The state-of-art models models (CNN, LSTM, RNN, GRUs) are unworthy for real time applications due to high number of parameters and high execution time. Therefore, our contribution is to propose a model as a tradeoff between the state-of-art models in terms of number of parameters, execution time, and accuracy. The proposed model is used to understand the sense of a real-life text conversation on a large data set with a high level of accuracy.

The basic model of text conversation is shown in the Fig. 1.

Figure 1 expresses different features that can be recognized through the text conversation. So, through a conversation, we can recognize the emotions, sentiments and sense. The Fig. 1 illustrate that sense understanding is absolutely dissimilar to all other features of conversation because it imitates the sense of conversation about which it is going on like "social & personal", "medical", "political", "educational", "advertisement", "games & sports", and "legal".

The remainder of this paper is organized as follows: Section 2 overviews the related research works and recent machine learning techniques proposed by various researchers from the same perspective. The proposed work is expounded in Section 3 and the detailed experimental results are shown in Section. 4. Finally, Section 5 provides conclusions and future scope.

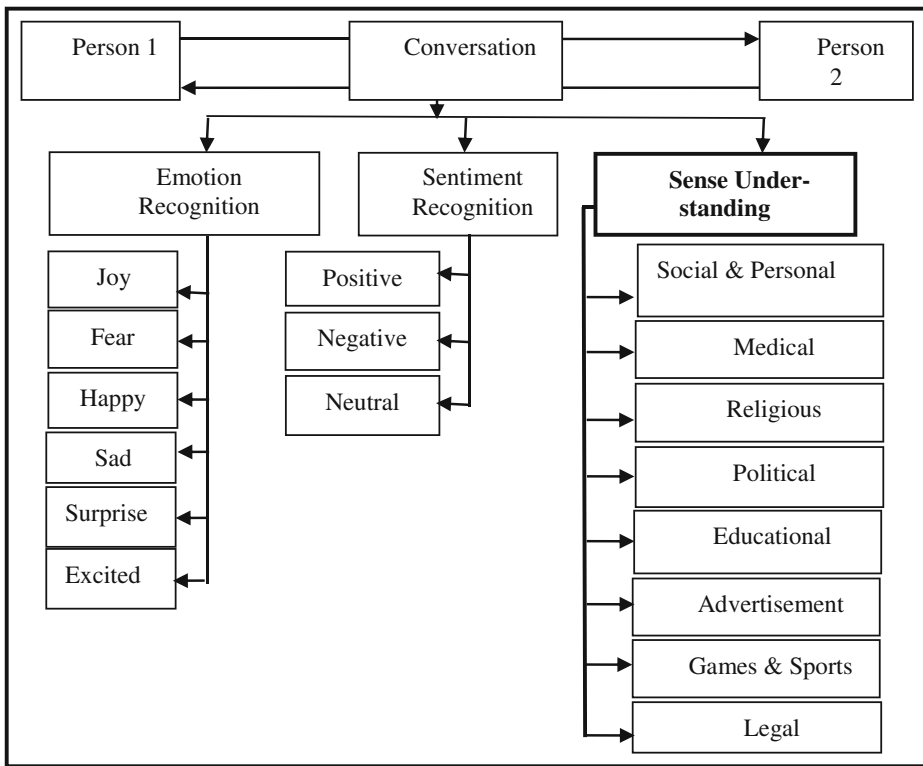


Fig. 1 The state of the art model of a Text conversation with its features

2 Related work

This section will explain the recent techniques and methods proposed by various researchers in the field of text processing and machine learning. In the existing scenario, advanced machine learning techniques like convolutional neural network (CNN), deep neural network (DNN), LSTM, and recurrent neural network (RNN) are commonly used for text processing. It can be utilized for recognizing the sentiments, emotions or moods, based on the text data. Despite of these aforesaid applications, the text can also be utilized to understand the sense of conversation about which a conversation has been taking place. Convolutional neural network and recurrent neural network are the most effective advance machine learning techniques for text mining and text processing [1].

Nowadays most of the companies are analyzing the customer’s feedback (collected through social media like Facebook, Twitter, WhatsApp, Mail, etc.) to increase their business. This analysis is utilized to increase the satisfaction level of their customers. Therefore, for this purpose, sentiment analysis on the customer’s data (reviews and comments) can be performed. The sentiment analysis of conversation using a multi fusion approach is proposed by [11]. Based on the customer feedback, three classes of sentiments are used in this paper. These sentiment classes of customer are positive customer, negative customer, and neutral customer. In their research, the author used Twitter’s data and also recognized the emotion of customers like happy, sad, and angry. Recently, most of the researchers perform sentiment analysis on Twitter’s data because of

its reality in nature [6]. The results of this paper indicate that CNN works very effectively to process the text, however, it used a high number of parameters to process. The paper is also limited to recognize the sentiments of customers only.

The fusion of both local and global contexts for enhancing the performance of sentiment analysis is proposed [14]. In this paper, the local context is used to assign the correct sense of a word in communication while the use of a global context is to define the contextual issues concomitant with a specific domain. In this paper, the weighted mean-based approach is used for obtaining the sentiment value of the customer. This proposed approach works with small size of the dataset with a baseline method. Moreover, to know the customer's behaviour or feedback about a specific product basis of conversation like tweets is proposed [18]. In this paper, a deep learning algorithm is used to perform the data extraction and sentiment analysis on specific products. The author also applied the convolution neural network to improve the accuracy of the proposed model. The result analysis shown in this paper indicates that CNN works more effectively to process the text, however, the paper is limited to recognize the sentiments of customers on a specific product only.

A text classification using a Multi-input convolutional neural network is proposed [10]. In this paper, the author executed pre-processing at two levels i.e. the word level and the character level. To validate the performance, the author used different datasets and compare the outcome with other classifiers also. However, the text classification obtained by the convolution neural network has higher accuracy than any of the other machine learning techniques. Therefore, it is clear by this paper that CNN is more appropriate for text classification and can be implemented to recognize the situation, emotion, or sense of a conversation. However, it used high number of parameters to process.

Multi-person activity recognition using a temporal convolution network is proposed [20]. In this paper, the author utilized motion trajectories to recognize the human activities by using two neural networks: long short-term memory (LSTM) network and a temporal convolutional network (TCN). The motion trajectory is classified into 15 classes of activities. The result analysis shown in this paper indicated that temporal convolution network outperformed. Moreover, a research on action segmentation is proposed [7]. In this paper, the author introduced a multi-stage architecture for action segmentation by using the temporal convolution network. Each stage includes a lot of expanded temporal convolutions to produce the prediction and it is refined by the next stage. The proposed model is trained using a combination of a classification loss that penalizes over-segmentation errors. The result analysis shows that the accuracy of the proposed model is satisfactory on different datasets. Therefore, the concept of temporal convolution network can also be applied to other field of research like sense recognition and classification.

Many researchers have already been proposed sentiment analysis on Twitter's data because it is famous social communication channel with large amount of data. Despite of being emotion, sentiment, event or activity and scene, sense of conversation can also be recognized through the communication. The objective of the paper is to recognize the sense of conversation using a temporal convolution neural network. Most of the researcher's applied the concept of temporal convolution neural networks for activity classification of a video data. However, the concept of temporal convolution neural network is not used in the text processing yet. Therefore, our objective is to use this novel approach for sense understanding in any communication using a large dataset.

Figure 2 shows the differentiation of state-of-art methods with the proposed method. CNNs are used to learn the local patterns. While learning, it considers the whole input at a

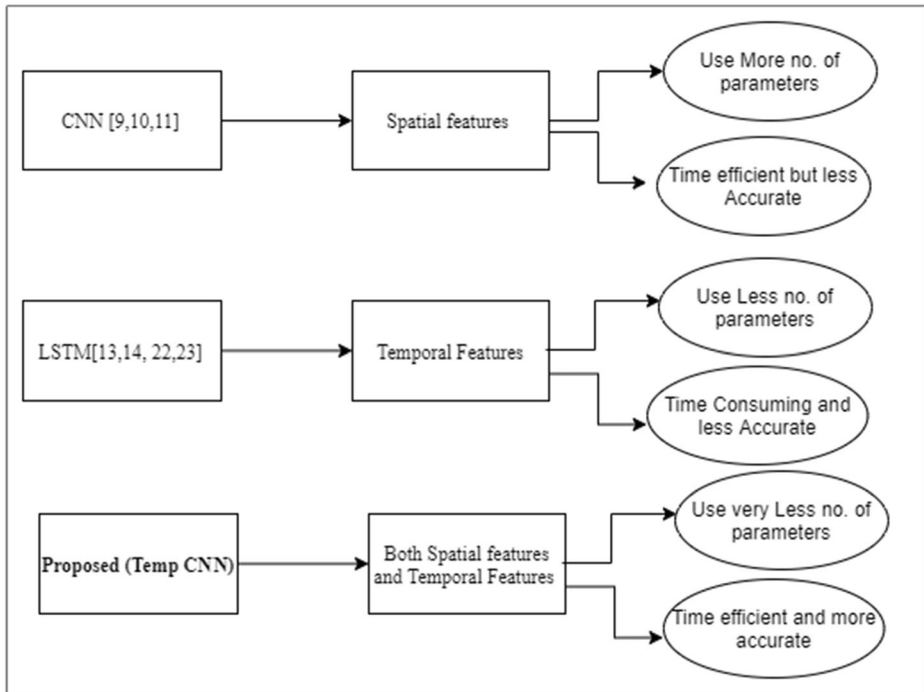


Fig. 2 Differentiation of state-of-the-art methods and the proposed one

time. It doesn't consider the temporal features i.e., the features from the time dependent sequences of the data. For example, the position of images in a video and position of words in a sentence and position of a sentence in a paragraph. This problem can be solved by using LSTM. They tend to learn the temporal features by considering the output of the previous cell vector for the calculation of current output. Thus, the research gap is to design such a model that can provide both the spatial and temporal features by reducing the total number of parameters.

3 Proposed work

We propose a Temporal CNN model for text classification. Instead of passing a single time-step data, here word, of a time-series data, here sentence, we pass the complete sentence to each Spatio Temporal cell. The current output is calculated on the basis of previous cell output, cell state from previous cell and the original input. Same process will happen N times. The value of N can vary according to the problem. This way each cell gets the same input data to process. Unlike in LSTM where each cell gets a different and a subset of a single instance our modified cell gets the same and full data each time.

The proposed model consists of different layers as shown in Fig. 2. The process of the proposed model starts with the preprocessing of a text conversation. Therefore, in preprocessing, each sentence is converted into a lowercase followed by tokenization and padding to length 65. Then these preprocessed sentences are passed to the embedding matrix. This embedding matrix's weights are obtained from training the model. The output of the

embedding layer is passed to the SpatioTemporal Neural network. The neural network contains the modified 10 sequentially connected SpatioTemporal cells. Each cell has nine shared one Dimensional CNN layers each with 16 filters and each filter of size 16. Additionally, we applied L2 regularization to avoid overfitting and with the same padding thus, resulting same input and output shape.

The SpatioTemporal cell takes the output from the embedding layer, the cell state, and output from the previous cell. The input vector is calculated by passing the input(X), previous state(H_{t-1}) to CNN layers W_{IX} , W_{IH} respectively. The individual outputs are added to the previous cell state(C_{t-1}) and finally the sigmoid activation (σ) is applied to it. The resultant vector is named as I . Mathematically; it can be represented as:

$$I = \sigma(W_{IX} * X + W_{IH} * H_{t-1} - 1 + C_{t-1}) \quad (1)$$

Here $*$ is the convolution operator.

Similarly, for the forget gate, the input(X), previous state(H_{t-1}) are passed to CNN layers W_{FX} , W_{FH} respectively. The individual outputs are added to the previous cell state(C_{t-1}) and finally the Sigmoid activation (σ) is applied to it. The resultant vector is named as F . Mathematically; it can be represented as:

$$F = \sigma(W_{FX} * X + W_{FH} * H_{t-1} - 1 + C_{t-1}) \quad (2)$$

Here $*$ is the convolution operator.

Then, the input(X), previous cell output(H_{t-1}) is passed to CNN layers W_{CX} , W_{CH} respectively. The resultant vectors are added and passed to the tanh activation function. The hadamard product of I and the resultant vector is added with the hadamard product of F and previous cell state(C_{t-1}). The resultant vector is named as C_t . Mathematically; it can be represented as:

$$C_t = F \circ C_{t-1} - 1 + I \circ \tanh(W_{CX} * X + W_{CH} * H_{t-1}) \quad (3)$$

Here $*$ is the convolution operator and \circ is the Hadamard product.

The output of the current cell is generated by passing the input(X), previous state(H_{t-1}) to CNN layers W_{OX} , W_{OH} respectively. The individual outputs are added with the current cell state(C_t) calculated above and finally the RELU activation is applied to it. The resultant vector is named as O (output). Mathematically, it can be represented as:

$$O = RELU(W_{OX} * X + W_{OH} * H_{t-1} + C_t) \quad (4)$$

Here $*$ is the convolution operator.

New state (H) for the current cell, the intermittent output which is passed to the next cell state is generated by Hadamard product between O (Eq. (4)) and tanh of current cell state (C_t). Mathematically, it can be represented as:

$$H_t = O \circ \tanh(C_t) \quad (5)$$

Here $*$ is the convolution operator and \circ is the Hadamard product.

The above process is repeated N times which will be called iterations. In each iteration, the weights of the CNN layers will remain shared.

The output from the last cell is flattened to a 1-dimensional vector and passed to an average pooling layer. This will reduce the output size to half. Then, the output of the Pooling layer is passed to a fully connected layer consisting of 128 units and Rectified Linear (Relu) as the activation function. The output of this layer is passed to a dropout layer followed by another fully connected layer with 8 units and softmax as the activation function. Finally, the index of maximum value present in the output array of FC(8) gives the final sense of the sentence.

Figure 3. shows the proposed model for sense understanding of text conversation while Fig. 4. shows the detailed working of SpatioTemporal cells.

3.1 Text pre-processing

The main advantage of pre-processed text is, it helps for reducing the vocabulary size and in turn the training time and space. The preprocessing is applied to the input text conversation

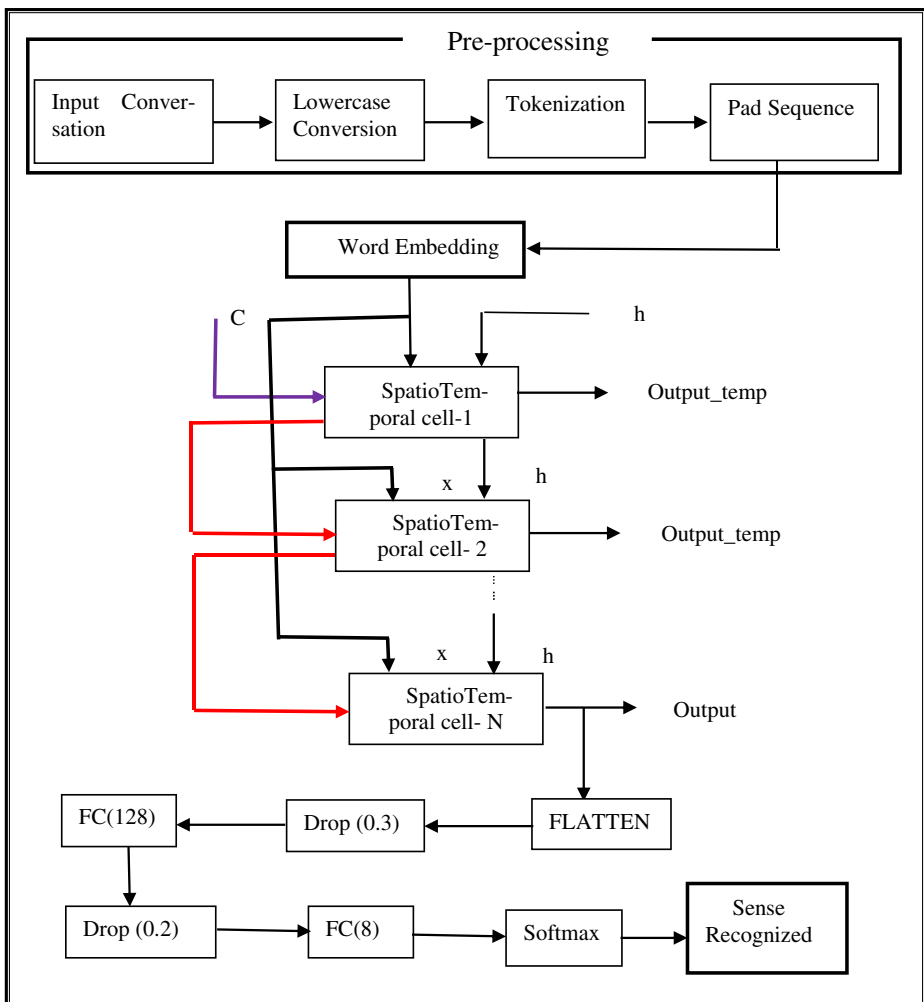


Fig. 3 Proposed framework of sense understanding

$$E = W.S \tag{6}$$

3.2 CNN

Convolutional neural network is a network that gives a feature array from an input array after applying convolution operation on the input array [16]. For this, several filters are convoluted on an input matrix. The output of each filter is concatenated and passed to the next layer. Mathematically it can be represented as:

$$CNN_i, d = \sum_{h=0}^H \sum_{w=0}^W (I_i + h, w * Kh, w), \forall i \leq N; d \leq D \tag{7}$$

Here, I is the input array of length N. The convolution is performed with D different kernels. K is the Kernel of shape [HxW].

3.3 Average pooling

An average pooling is a method to minimize the variance in an input matrix. It down-samples the input matrix. In average pooling, the output matrix is derived by taking the average of a moving window on the input matrix. In our proposed algorithm, the 1D average pooling is used to down-sample the input matrix to its half of the input size with 2 as the kernel size and stride as 2. Mathematically, it can be represented as:

$$AP = \frac{1}{2} \sum_{w=0}^2 I_{i+w} \text{ for } \forall i \in \{2 * k \forall k : 0 \text{ to } N/2\} \tag{8}$$

Here, I is the input array of length N.

3.4 FC

A Fully Connected layer (FC) finds new features from an existing set of features using a weight matrix and bias. A fully connected layer uses a non-linear activation function. Technically, it is the product of weight matrix W and input image I. The dimension of W is [NxO], where N is the size of the input array and O is the output shape. Afterwards, bias (B) is added to the intermittent results. Array B is of size [O]. Activation functions such as sigmoid, RELU can be used to introduce non-linearity into the layer. It can be represented by Eq. (9) as:

$$R = W.I + B \sim \tag{9}$$

In the proposed model, the first dense layer with 128 units, Rectified Linear Unit (ReLU) is used as the activation function. Then the output from this layer is passed to the second fully connected layer FC(8) with Softmax as the activation function, which yields the final probabilities for eight senses of the proposed model.

The complete working of the proposed model can also be represented by an algorithm as:
Proposed training algorithm:

Input: E(embedding matrix), Actual_Predictions, Training_Dataset

Output: Trained model

1. *Initialize:*
 - a. C_{t-1} = Zero Matrix
 - b. H_{t-1} = Zero Matrix
2. *For each instance i in training set:*
 - a. Il = Lowercase(i)
 - b. It = Tokenize(Il)
 - c. Ip = padding(It)
 - d. $I.append(Ip)$
3. *For each batch of training data:*
 - a. $I_e = E.I$
 - b. $W_{IX} = CNN(filters=16, kernel=16, regularizer = 'l2', pad='same')$
 - c. $W_{IH} = CNN(filters=16, kernel=16, regularizer = 'l2', pad='same')$
 - d. $W_{FX} = CNN(filters=16, kernel=16, regularizer = 'l2', pad='same')$
 - e. $W_{FH} = CNN(filters=16, kernel=16, regularizer = 'l2', pad='same')$
 - f. $W_{CX} = CNN(filters=16, kernel=16, regularizer = 'l2', pad='same')$
 - g. $W_{CH} = CNN(filters=16, kernel=16, regularizer = 'l2', pad='same')$
 - h. $W_{OX} = CNN(filters=16, kernel=16, regularizer = 'l2', pad='same')$
 - i. $W_{OH} = CNN(filters=16, kernel=16, regularizer = 'l2', pad='same')$
 - j. *Loop for 10 times:*
 - i. $I = Sigmoid(W_{IX}(I_e) + W_{IH}(H_{t-1}) + C_{t-1})$
 - ii. $F = Sigmoid(W_{FX}(I_e) + W_{FH}(H_{t-1}) + C_{t-1})$
 - iii. $C_t = F \circ C_{t-1} + I \circ \tanh(W_{CX}(I_e) + W_{CH}(H_{t-1}))$
 - iv. $O = RELU(W_{OX}(I_e) + W_{OH}(H_{t-1}) + C_t)$
 - v. $H_t = O \circ \tanh(C_t)$
 - k. $F = Flatten(O)$
 - l. $fc1 = FullyConnected(units = 128, activation = Relu, input = F)$
 - m. $Do1 = Dropout(0.2, input = fc1)$
 - n. $fc2 = FullyConnected(units = 8, activation = Relu, input = do1)$
 - o. $Os = Softmax(fc2)$
 - p. $Om = Argmax(Os)$
 - q. $error = Categorical\ Cross\ Entropy(Om, actual\ predictions)$
 - r. *Back Propagate Error(error)*
4. *Return trained model*

Proposed testing algorithm:*Input: Training set, Trained model,**Output: Sense_predictions*

1. For each instance i in training set:
 - a. $I_l = \text{Lowercase}(i)$
 - b. $I_t = \text{Tokenize}(I_l)$
 - c. $I_p = \text{padding}(I_t)$
 - d. $I.append(I_p)$
2. $I_e = E.I$
3. $W_{IX} = \text{CNN}(\text{filters}=16, \text{kernel}=16, \text{regularizer} = 'l2', \text{pad}='same')$
4. $W_{IH} = \text{CNN}(\text{filters}=16, \text{kernel}=16, \text{regularizer} = 'l2', \text{pad}='same')$
5. $W_{FX} = \text{CNN}(\text{filters}=16, \text{kernel}=16, \text{regularizer} = 'l2', \text{pad}='same')$
6. $W_{FH} = \text{CNN}(\text{filters}=16, \text{kernel}=16, \text{regularizer} = 'l2', \text{pad}='same')$
7. $W_{CX} = \text{CNN}(\text{filters}=16, \text{kernel}=16, \text{regularizer} = 'l2', \text{pad}='same')$
8. $W_{CH} = \text{CNN}(\text{filters}=16, \text{kernel}=16, \text{regularizer} = 'l2', \text{pad}='same')$
9. $W_{OX} = \text{CNN}(\text{filters}=16, \text{kernel}=16, \text{regularizer} = 'l2', \text{pad}='same')$
10. $W_{OH} = \text{CNN}(\text{filters}=16, \text{kernel}=16, \text{regularizer} = 'l2', \text{pad}='same')$
11. Loop for 10 times:
 - a. $I = \text{Sigmoid}(W_{IX}(I_e) + W_{IH}(H_{t-1}) + C_{t-1})$
 - b. $F = \text{Sigmoid}(W_{FX}(I_e) + W_{FH}(H_{t-1}) + C_{t-1})$
 - c. $C_t = F \circ C_{t-1} + I \circ \tanh(W_{CX}(I_e) + W_{CH}(H_{t-1}))$
 - d. $O = \text{RELU}(W_{OX}(I_e) + W_{OH}(H_{t-1}) + C_t)$
 - e. $H_t = O \circ \tanh(C_t)$
12. $F = \text{Flatten}(O)$
13. $fc1 = \text{FullyConnected}(\text{units} = 128, \text{activation} = \text{Relu}, \text{input} = F)$
14. $Do1 = \text{Dropout}(0.2, \text{input} = fc1)$
15. $fc2 = \text{FullyConnected}(\text{units} = 8, \text{activation} = \text{Relu}, \text{input} = do1)$
16. $Os = \text{Softmax}(fc2)$
17. $Om = \text{Argmax}(Os)$
18. Return *Sense_predictions*

4 Experimental results

This section presents experimental results of the proposed model on the large human-annotated dataset. To validate the performance of the proposed model, confusion matrices and measure parameters (like recall, precision, f-score) are also represented.

4.1 Experimental setup

The proposed model is executed on Python 3.7 programming language with IDE Jupyter Notebook 6.0, Intel i5 processor with 16 GigaBytes RAM. To create a large corpus dataset, we have used various available standard open text datasets, conversations, essays available online, news articles, etc. These sentences have an average length of 30 words. 70% of the sentences belong to one of the target categories and the rest of the 30% can be classified to multiple outputs. Each sentence is then manually tagged to one of the eight senses.

For the purpose of model evaluation, to understand the sense of a conversation, we have taken total instances 180,354. Out of these, 150,179 instances are used for training and 30,175 instances are used for testing the performance of the proposed model. The detailed individual category wise instances used in the proposed model are shown in the Table 1. While training, we used Adam as the optimizer with initial learning rate as 1e-4 and it gets reduced each time on plateaus.

We have also used a standard dataset, IMDB sentiment classification dataset [4], for checking the efficiency and performance of the proposed model. We used 25,000 sentences for training and another 25,000 for validation and testing. Vocabulary size taken is 20,000 words. The average length of sentences is 1000.

4.2 Results on training and test data

To validate the performance of the proposed model, we check the performance on training & test data. Various measures such as classification accuracy, confusion matrix and the true predictions on training data are represented by the confusion matrix [12, 13]. A confusion matrix is a procedure for summarizing the performance of sense classification as shown in Fig. 5. The accuracy of the proposed model can be calculated as:

$$Accuracy = \frac{\sum \text{True Predications}}{\text{Total Instances}} \quad (10)$$

Figure 5 shows the confusion matrix of training results. It represents the true positive, true negative, false negative and false positive of the individual sense category. In the figure, the true predictions are 148,600 and total instances are 150,179. Therefore, the training accuracy of the proposed model (by using Eq. (10)) is 98.94%.

Table 1 Number of instances used for training and testing

Domain category	Training	Testing
Religious	18,894	3790
Medical	18,940	3805
Advertisement	18,554	3722
Political	19,530	3926
Personal/ Social	18,473	3710
Education	18,573	3734
Games & Sports	18,783	3771
Legal	18,442	3717
Total	150,179	30,175

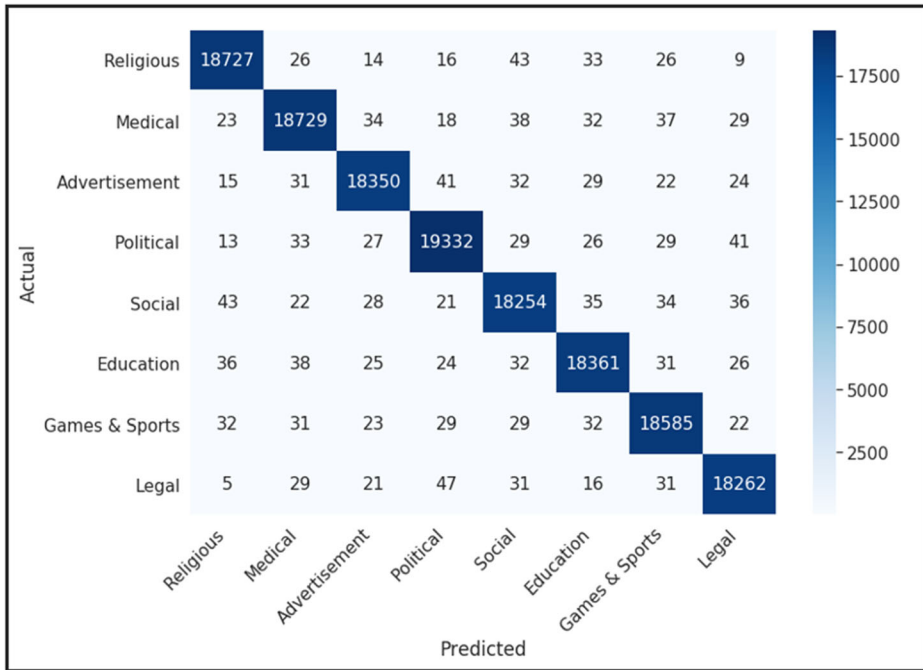


Fig. 5 Training results of the proposed model

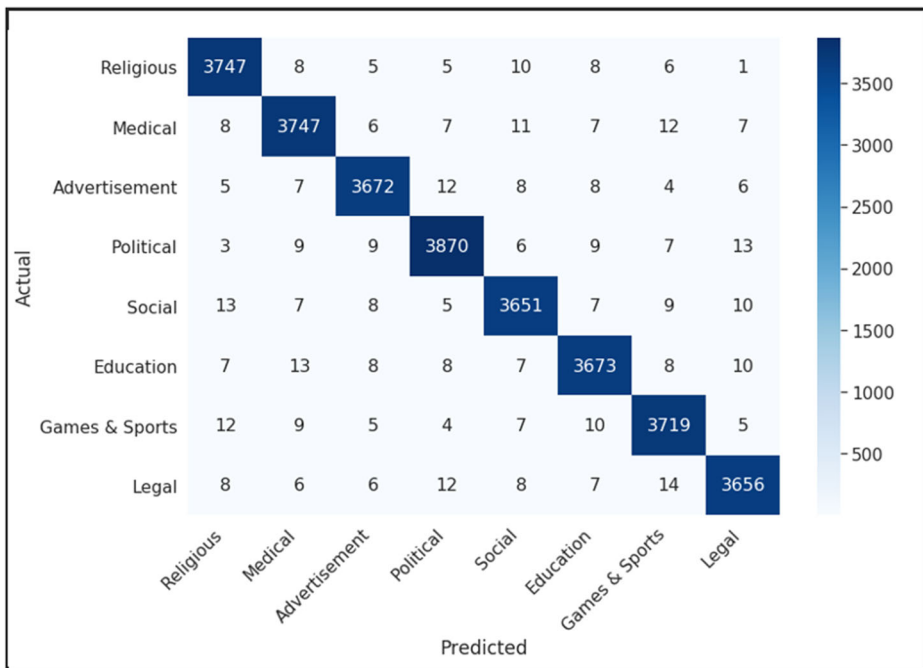


Fig. 6 Testing results of the proposed model

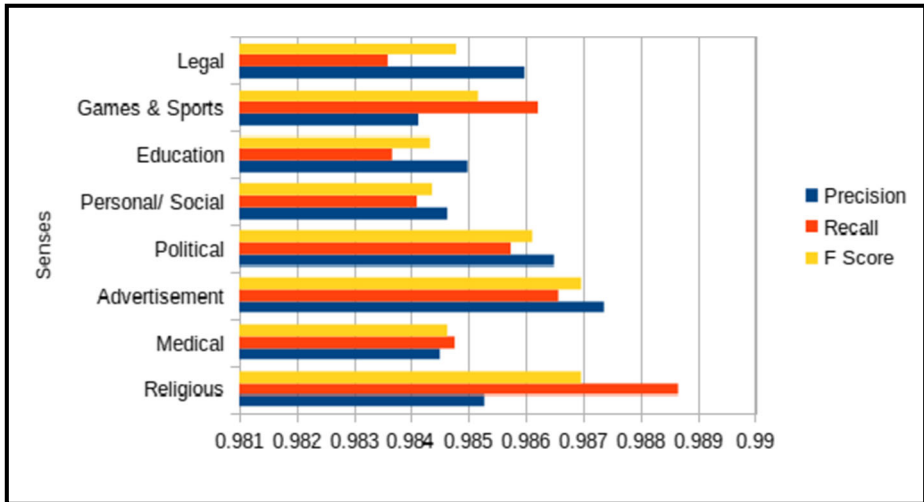


Fig. 7 Measure parameters (Precision, recall, and F-score values) on the validation set

The proposed model is applied to the test data set and the performance is shown in Fig. 6 by using a confusion matrix. In figure, the true predictions are 29,735 and total instances are 30,175. Therefore, the test accuracy of the proposed model can be obtained by using Eq. (10), i.e. 98.54%.

Figure 7 shows the performance of the proposed model on custom made sense classification dataset. It shows the precision, recall and F-score obtained on different Senses, each scoring a value above 0.984. The three parameters are calculated using the one vs. all strategy.

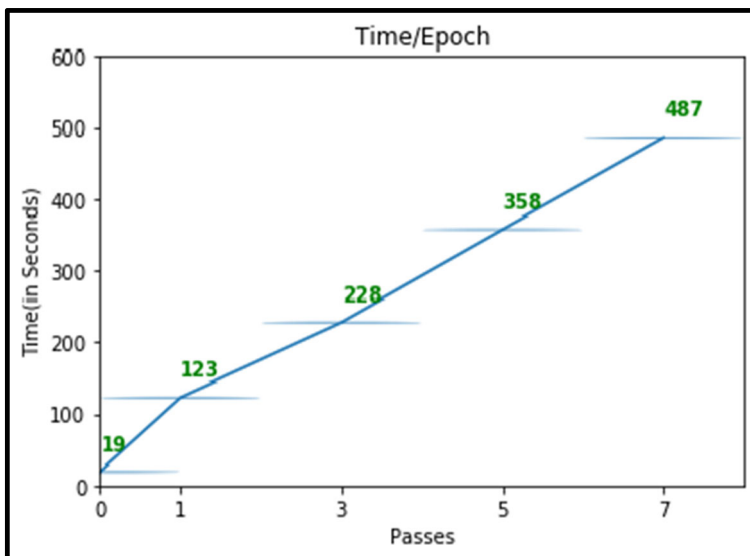


Fig. 8 Execution time per Epoch on a different number of iterations

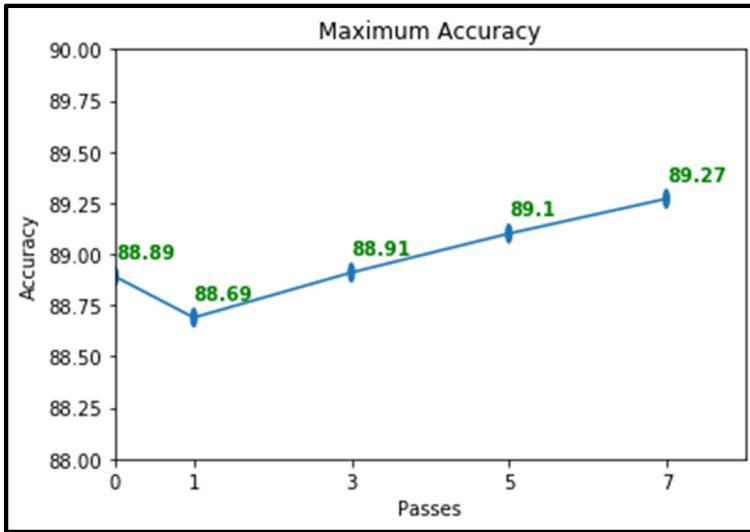


Fig. 9 Accuracy of a different number of iterations

4.3 Performance on varying iterations

To check the effect of the number of iterations on the final validation accuracy we executed the model with different number of iterations. We choose 0, 1, 3, 5, and 7 as the number of iterations. Exploration of higher iterations are limited due to the resources. Figure 8 shows the execution time as we increase the number of iterations keeping the total number of parameters of the model remains constant to 1,160,754. It is clear that as the number of iterations increased, the execution time per epoch also increases.

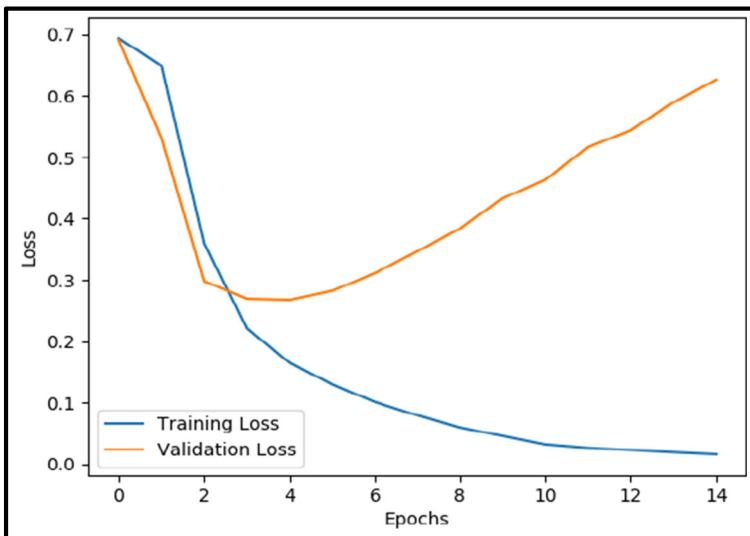


Fig. 10 Training and validation loss of proposed model on IMDB dataset

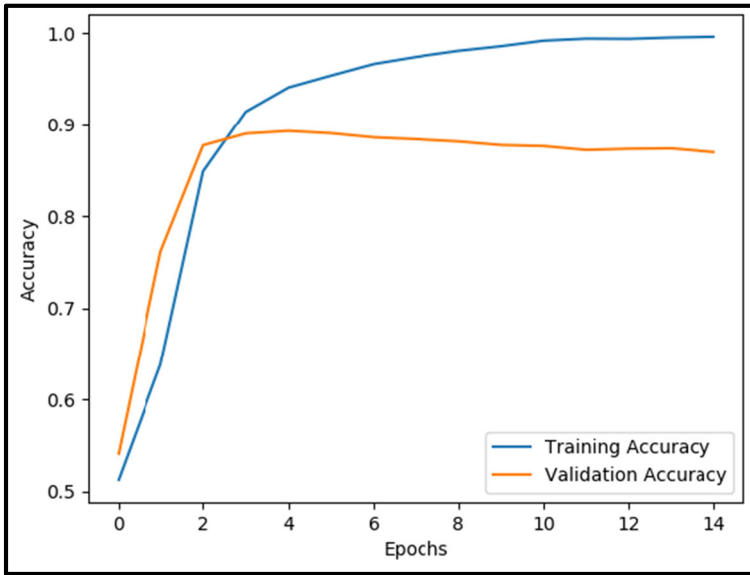


Fig. 11 Training and validation loss of proposed model on IMDB dataset

It is clear from Fig. 9 that, as the number of iterations increases the accuracy of the model increases.

4.4 Model performance on IMDB sentiment analysis dataset

To check and validate our model on a standard dataset, we executed our proposed model using the IMDB sentiment dataset. For this, we used the vocabulary size of 20,000 i.e., the total number of different words used is 20,000. Each sentence is padded to a length of 1000. The embedding layer generated the feature vector for each word of length 32. The model has a total

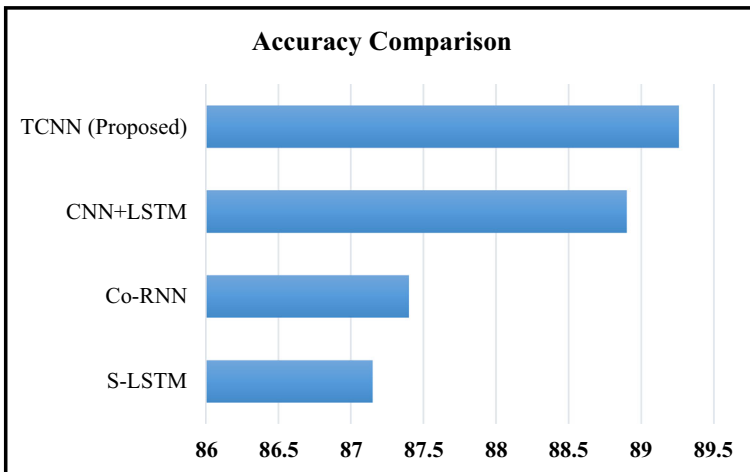


Fig. 12 Comparison of the proposed model with similar models on IMDB dataset [23]

Table 2 Parameters comparison of the proposed model with different existing models

Parameters / Models	Bi-LSTM	CNN	Proposed Temporal CNN
Trainable parameters	389,365	2,163,605	1,160,754
Accuracy	88.34	88.81	89.27
Time (in Minute)	37	2	8

of 1,160,754 parameters. The model is trained on 25,000 sentences and further evaluated on 25,000 sentences. The number of iterations was kept to 7.

It is evident from the Figs. 10 and 11 that the model started over-fitting after the fifth training iteration.

Figure 12, shows that the proposed model is better than the LSTM [8, 17], CNN, LSTM + CNN models [2, 15].

Table 2 shows the comparison between the proposed model and two benchmark models (CNN & BiLSTM) on IMDB sentiment classification dataset. The proposed algorithm has the best accuracy as compared to CNN and Bi-LSTM. The drawback with a Bi-LSTM model is high execution time which makes it slower and unworthy for real time applications. This problem has been resolved by the proposed model as it takes only 8 min per epoch on a batch size of 256. The drawback with the CNN model is the high number of parameters and in turn high memory(RAM) requirement. Our proposed model used half number of the parameters in compared to CNN model. This way our proposed model is a tradeoff between these two models in terms of number of parameters and execution time while providing higher accuracy.

5 Conclusions and future scope

In this paper, we have proposed a model to understand the sense of text conversation by using temporal convolution technique. The proposed model outperforms with an accuracy of 98.54%. The proposed model is best suited to understand the sense of real conversation and can be promoted to implement human-computer interaction applications. The proposed model is based on advanced machine learning techniques. The performance comparison of the proposed model indicates that it is better than other state-of-art models in all respects. The number of iterations was limited to 7 due to resource constraints. The efficiency of more iterations can be explored in future. The sense understanding can also be applied to audio communication in the near future.

Declarations

Conflict of interest All authors declare that they have no conflict of interest.

References

1. Banerjee I, Ling Y, Chen MC, Hasan SA, Langlotz CP, Moradzadeh N, ... Farri O (2019) Comparative effectiveness of convolutional neuralnetwork (CNN) and recurrent neural network (RNN) architectures for radiologytext report classification. *Artif Intell Med* 97:79-88

2. Camacho-Collados J, Pilehvar MT (2017) On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis. arXiv preprint arXiv:1707.01780
3. Feizollah A, Amin S, Anuar NB, Abdullah NAB, Hazim M (2019) Halal products on Twitter: Data extraction and sentiment analysis using stack of deep learning algorithms. *IEEE Access* 7:83354–83362
4. Gui L, Jia L, Zhou J, Xu R, He Y (2020) Multi-task learning with mutual learning for joint sentiment classification and topic detection. *IEEE Transactions on Knowledge and Data Engineering*. <https://doi.org/10.1109/TKDE.2020.2999489>
5. Hall JK (2019) The contributions of conversation analysis and interactional linguistics to a usage-based understanding of language: Expanding the transdisciplinary framework. *Mod Lang J* 103:80–94
6. Jianqiang Z, Xiaolin G, Xuejun Z (2018) Deep convolution neural networks for twitter sentiment analysis. *IEEE Access* 6:23253–23260
7. Li SJ, AbuFarha Y, Liu Y, Cheng MM, Gall J (2020) Ms-tcn++: Multi-stage temporal convolutional network for action segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. <https://doi.org/10.1109/TPAMI.2020.3021756>
8. Liu G, Guo J (2019) Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing* 337:325–338
9. Luo LX (2019) Network text sentiment analysis method combining LDA text representation and GRU-CNN. *Personal Uniquit Comput* 23(3–4):405–412
10. Parчета Z, Sanchis-Trilles G, Casacuberta F, Redahl R (2019) Multi-input CNN for text classification in commercial scenarios. In: *International Work-Conference on Artificial Neural Networks*. Springer, Cham, pp 596–608
11. Poonpakdee P, Koiwanit J, Yuangyai C (2017) Decentralized network building change in large manufacturing companies towards Industry 4.0. *Procedia Comput Sci* 110:46–53
12. Rathor S, Jadon RS (2019) Acoustic domain classification and recognition through ensemble based multilevel classification. *J Ambient Intell Humaniz Comput* 10(9):3617–3627
13. Rathor S, Jadon RS (2019) The art of domain classification and recognition for text conversation using support vector classifier. *Int J Arts Technol* 11(3):309–324
14. Rintyarna BS, Sarno R, Faticah C (2020) Enhancing the performance of sentiment analysis task on product reviews by handling both local and global context. *Int J Inf Decis Sci* 12(1):75–101
15. Rusch TK, Mishra S (2020) Coupled Oscillatory Recurrent Neural Network (coRNN): An accurate and (gradient) stable architecture for learning long time dependencies. arXiv preprint arXiv:2010.00951
16. Sharma AK, Chaurasia S, Srivastava DK (2020) Sentimental short sentences classification by using CNN deep learning model with fine tuned Word2Vec. *Procedia Comput Sci* 167:1139–1147
17. Shi X, Chen Z, Wang H, Yeung DY, Wong WK, Woo WC (2015) Convolutional LSTM network: A machine learning approach for precipitation nowcasting. arXiv preprint arXiv:1506.04214
18. Voramontri D, Klieb L (2019) Impact of social media on consumer behaviour. *Int J Inf Decis Sci* 11(3): 209–233
19. Wang X, Jiang W, Luo Z (2016) Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In: *Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers*, pp 2428–2437
20. Wang J, Du Z, Li A, Wang Y (2020) Assessing action quality via attentive spatio-temporal convolutional networks. In: *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*. Springer, Cham, pp 3–16
21. Widiastuti NI (2019) Convolution neural network for text mining and natural language processing. In: *IOP Conference Series: Materials Science and Engineering*, vol 662, no 5. IOP Publishing, Bristol, p 052010
22. Yadollahi A, Shahraki AG, Zaiane OR (2017) Current state of text sentiment analysis from opinion to emotion mining. *ACM Comput Surv (CSUR)* 50(2):1–33
23. Zhang Y, Liu Q, Song L (2018) Sentence-state lstm for text representation. arXiv preprint arXiv:1805.02474

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.