# Recurrent Convolutional Neural Networks for Speech Act Recognition

**Anand P. Popat**
Department of Computer Science
University at Buffalo
Buffalo, NY 14226
apopat@buffalo.edu

**Mugdha M. Ansarwadekar**
Department of Computer Science
University at Buffalo
Buffalo, NY 14226
mugdhami@buffalo.edu

## Abstract

Spoken Language Understanding is one of the most important aspect of Natural Language Processing especially in dialogue system. It's a challenging problem which aims at characterizing the semantic labels to utterances of the dialogue system. We have performed extensive experiments on a major benchmark dataset Switchboard Dialogue Act (SWDA) which involves SVM, CNN and our proposed models consisting of Recurrent Convolutional Neural Networks to achieve results which are on par with the state-of-the-art systems. We conducted several experiments to perfectly tune the hyper-parameters through which state-of-the-art accuracy is obtained with just one epoch using one CPU which is remarkable.

## 1 Introduction

Spoken Language Understanding is a fundamental task which has its applications in speech recognition, text generation, machine translation and semantic analysis. Several proposals have been made on Switchboard Dialogue Act (SWDA), AMI, MRDA, DSTC-4 and DSTC-5. As described in Chen et al. [1], we conducted experiments on basic machine learning techniques as a ground result. However, these techniques lack contextual information and cannot capture dependencies between utterances in a dialogue.

Deep Learning models have achieved remarkable success recently in Spoken Language Understanding (SLU) after the proposed model using Convolutional Neural Networks described in Kim., 2014 [2]. Though the model has a breakthrough performance, it has some limitations such as the use of pre-trained work vectors which increases the overhead in training and though it captures the inter-word dependencies in a sentence, it still lacks the contextual information between the sentences.

In this work, we propose several models which leverage the inter-sentence information through Recurrent Convolutional Neural Networks. Unlike previous work (Xu et al[3], Kim.[2]), our models do not use pre-trained word vectors like GLoVe or Google's word2vec which decreases the computational time with no impact on the accuracy. Also unlike Ushio et al[4], our models do not perform Multi-label classification but are concentrated on finding inter-sentence dependencies for single label classification achieving better results for single label classification.

To summarize, our contributions are as follows:

- We present a variety of Recurrent Convolutional Neural network models with an achieved accuracy of 70.77% with our prime model.
- Lesser parameters and fine tuning of the parameters than the previous works which significantly reduces the training time period.
- A comprehensive study of different methods used so far for speech act recognition problem and a comparison to our proposed models

The remained of the paper is organized as follows. In Section 2, we define all the different models along with the proposed models. In Section 3, we give an insight on the dataset and the experiment setup. In Section 4, we show the results that we got using different hyperparameter tuning. In Section 5, we discuss the performance of the previous models with our proposed models and in Section 6, we provide the conclusion along with future works.

# 2 Model

## 2.1 Basic ML

Basic Machine Learning techniques are used to produce the ground result like Logistic Regression, Decision Tree Classifier, Random Forest Classifier and SVM. The data is pre-processed for that and the features are designed manually. The following features were used: Bag Of Words (BOW) , Question Mark, WH-question, I Don't Know words, Yes-Words, No-Words, Do-Words, Non-Verbal Count, Apology Words, Thanking Words, UH-Count, Garzon Feature set. We have used scikit-learn to implement different algorithms and evaluate the model with different hyperparameters.

## 2.2 Convolutional Neural Network

Modifications to the previous design and the one described in Kim.,2014[2], we designed a model based on Convolutional Neural Network which has been very successful in Computer Vision and also in NLP to get semantic analysis of sentences. The model is shown in Fig. 1, let $x_i \in \mathbf{R}^k$ where its a *k*-dimensional word vector corresponding to the i-th word. A sentence of length *n* is pre-padded, post-padded or is of fixed length like median of all the sentences and is represented as

$$X_{1:n} = X_1 \oplus X_2 \oplus \dots \oplus X_n,$$

where $\oplus$ is the concatenation operator.

A filter window $\mathbf{w} \in \mathbf{R}^{hk}$ is applied to a window of *h* words to produce a feature map. The feature extracted would be

$$c_i = f(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b),$$

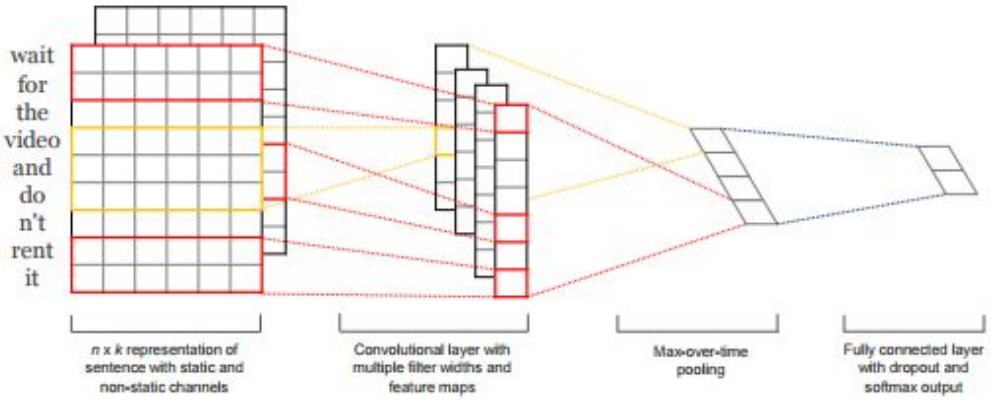
where *b* is the bias term and *f* is a nonlinear function like ReLU or tanh. The possible filter map obtained after applying this filter with different window sizes equates to

$$\mathbf{c} = [c_1, c_2, \dots, c_{n-h+1}],$$

We then apply GlobalMaxPooling on this to obtain the maximum value out of the feature set $C = \max\{\mathbf{c}\}$. This is important to have fixed sized feature map for the next fully connected layer.

The output of the max-pooling is passed through a fully-connected layer with dropout and softmax output.

As a modification to this model, we are using just one convolutional layer with different window sizes *h,* merging the feature sets of different window sizes and then passing it through Dense (fully connected layer). The hyperparameter tuning and results are described in Section 4.




## 2.3 Recurrent Neural Network

Long short-term Memory (LSTM) which is a type of Recurrent Neural Network has a vector of memory cell $c_t \in \mathbf{R}^h$ and a set of gates for storing and forgetting information used inside its networks. The architecture used for LSTM performs using the following equations.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\mathbf{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$
$$C_t = f_t * C_{t-1} + i_t * \mathbf{C}_t$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * tanh(C_t)$$

where *f* is the forget gate, *i* is the input gate, ***C*** is the vector of new candidate values, *C* is the new cell values to be remembered and *o* is the output gate.

LSTM is used to store information about the utterance tag of the previous word/sentence and predict the outcome of the next sentence. We are using two LSTM layers with different sets of hyperparameters described in Section 4.

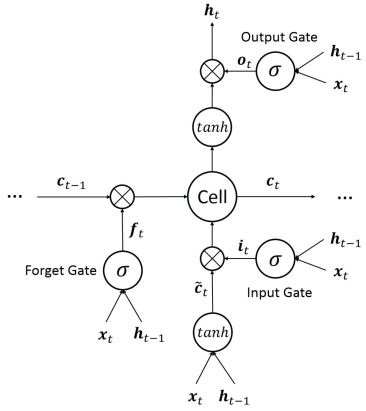



**Fig. 1: Structure of LSTM unit cell**

## 2.4 Recurrent Convolutional Neural Network

In this model, the input sentences are embedded using the embedding layer and fed to a Convolutional Neural Network. The output of the global max pooling i.e the feature maps from the

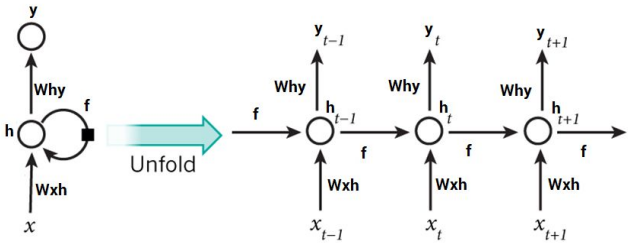



**Fig. 2: Recurrent Neural Network Architecture**

CNN are then fed to the LSTM as an input.

As a result the features between the words of the sentences is extracted by CNN and based on the previous word of the sentence, LSTM predicts the current word. Tuning of the hyperparameters is described in Section 4.

## 2.5 Recurrent Neural Network for Sentences

In this model, the only thing that is different from the model described in Section 2.3 is that instead of a single sentence being fed to the LSTM layer, we are now passing bunch of sentences to the LSTM so that the output tag of the previous sentences is remembered by the LSTM. The depth of the LSTM varies in the experimentation but it is crucial as the prediction of the dialogue act of the current utterances is dependent on the previous *d* utterances. Here each vector is given by different weightings in the sentence and sequence model.

$$h_t^{attr} = LSTM(s_t^{attr}, h_{t-1}^{attr})$$

## 2.6 Recurrent Convolutional Neural Network for Sentences

This model is an extension of the model described in Section 2.4. The difference between the two models is that in this model, the input to the CNN is a bunch of statements (amount may vary) and each input statement is separately processed through the CNN layer.

Once we have the output of the CNN layer, we merge the sentences together and feed it as in input to the LSTM. In this way, the dialogue tag of the previous *d* sentences will be remembered by the LSTM and an optimum result will be produced. In fact, this model has the best performance out of all the proposed models.

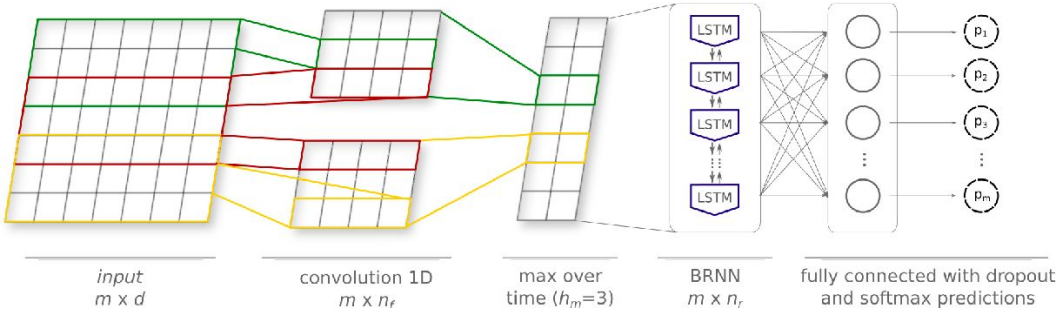



**Fig. 3: Recurrent Convolutional Neural Network Architecture**

## 3 Experimental Setup

For experiments, we are using Switchboard Dialogue Act (SWDA) corpus. The dataset consists of 1155 5-minute conversations, comprising of 2,05,000 utterances and 1.4 billion words. There are 220 tags used for coding and the frequency of occurrences of all the tags was studied. Based on this, these tags were clustered into 46 larger classes. This dataset is chosen as it is the most widely used scheme for studying human-to-human dialogues with two speakers.

The statistics of the dataset are given in Table 1 below.

| | M | D | L | C | W |
|---|---|---|---|---|---|
| Train | 178215.0 | 18378.0 | 45.0 | 1.0 | 8.482782 |
| Test | 44553.0 | 10210.0 | 44.0 | 1.0 | 8.487352 |

**Table 1: M: Number of Utterances, D: Size of Vocabulary, L: Size of label set, C: Average number of labels per utterance, M: Average length of Utterance**

Table 2 shows the type of utterances in the chat corpus..

| Tag | Example | proportion |
|---|---|---|
| STATEMENT | "I am working on my projects trying to graduate." | 36% |
| BACKCHANNEL/ACKNOWLEDGE | "Uh-huh." "Yeah." "All right." "Ok..." "Well..." | 19% |
| OPINION | "I think it's great." / "I don't believe it can work." | 13% |
| ABANDONED/UNINTERPRETABLE | "So, -" "Are yo-" "Maybe-" | 6% |
| AGREEMENT/ACCEPT | "That's exactly it." "I can't agree more." | 5% |

**Table 2: Top five percentages of utterance type in the SWDA corpus**

| | |
|---|---|
| Acknowledge | Apology |
| Declarative Wh-Ques | Hedge |
| No Answers | Non-Verbal |
| Self-Talk | St-Non-opinion |
| Statement-opinion | Wh-Questions |
| Yes-Answers | Yes-No-Ques |
| No-Answers | Rest |

**Table 3: Definition of attributes used for classification.**

## 4 Results and Discussion

In order to consider the relationship between the time distributed samples of the conversation data, we started with the implementation of Recurrent Neural Network (RNN). However, Vanilla RNN have poor performance compared Long Short Term Memory Units (LSTM) and Gated Recurrent Units (GRU). This guided us to consider both LSTM and GRU, which advanced implementations of Vanilla RNNs, to evaluate our models based on time series data. We have implemented 4 models incorporating the advantages of Convolutional Neural Networks and Recurrent Neural

Networks. This section discusses the performance of these model on our dataset for various settings of hyperparameters.

| Accuracy | Embedding | Embedding Dim | Word Length |
|---|---|---|---|
| 69.39 | Keras Layer | 300 | 81 |
| 69.95 | Keras Layer | 200 | 81 |
| 65.67 | Keras Layer | 300 | 6 |
| 64.3 | Keras Layer | 200 | 6 |
| 68.95 | Glove | 300 | 81 |
| 67.08 | Glove | 200 | 81 |
| 66.35 | Glove | 300 | 6 |
| 64.28 | Glove | 200 | 6 |

**Table 4 : Performance of Recurrent Neural Network Sentence based on sentence formatting**

| Accuracy | Embedding | Embedding Dim | Word Length |
|---|---|---|---|
| 70.55 | Keras Layer | 300 | 81 |
| 66.15 | Keras Layer | 300 | 6 |
| 70.25 | Glove | 300 | 81 |
| 66.35 | Glove | 300 | 6 |

**Table 5 : Performance of RCNN Sentence based on sentence formatting**

Recurrent Neural Network architecture implemented using LSTM and GRU performs well by considering the time relation between words of a single sentence. This architecture is able to extract the word features and accuracy enhances further when incorporated with Convolutional Neural Network architecture. RCNN architecture is able to extract latent features and consider the temporal relationship between words of a sentences.

As our dataset mainly consists of conversation, we decided to consider the temporal relationship between a group of sentences. This led to the two architectures described in sections 2.5 and 2.6. We have experiments by varying the number of sentences considered a block and our results prove that the ideal number of sentences to be considered for temporal relationship are four for the mentioned dataset.

We have observed a significant impact of sentence length, type of padding used for sentences and method of embedding the words on the performance of the models described above. Considering the maximum length of the sentence by post-padding shorter sentences gives the optimal performance. On the other hand, when the median length of the sentence is considered, a significant portion of the data is lost which results in poor performance of the model. Further, we

experimented by varying the method to embed the words, embedding using Glove library and embedding using Keras Embedding layer. The observed results concluded that the model performed better with Keras Embedding layer.

| Architecture | RNN Variant | Sentence Length | CNN filter size | Accuracy |
|---|---|---|---|---|
| RNN Sentence | LSTM | 81 | - | 69.39 |
| RNN Sentence | GRU | 81 | - | 69.53 |
| RNN Sentence | LSTM | 6 | - | 65.67 |
| RNN Sentence | GRU | 6 | - | 64.46 |
| RCNN Sentence | LSTM | 81 | 200 | 70.55 |
| RCNN Sentence | LSTM | 81 | 150 | 70.77 |
| RCNN Sentence | GRU | 81 | 200 | 69.96 |
| RCNN Sentence | GRU | 81 | 150 | 69.28 |

**Table 6: Comparison of LSTM and GRU**

| Model | Accuracy | Hyperparameters | | | |
|---|---|---|---|---|---|
| | | Sentence Length | Sentence Padding | Number of Sentences | Embedding |
| RNN | 70.31 | 81 | Pre-Padding | - | Keras Layer |
| RCNN | 68.21 | 81 | Pre-Padding | - | Keras Layer |
| RNN Sentences | 69.53 | 81 | Pre-Padding | 4 | Keras Layer |
| RCNN Sentences | 70.77 | 81 | Pre-Padding | 4 | Keras Layer |

**Table 7 : Comparison of architectures with optimal hyperparameters**

## 6 Conclusion

In this paper, we describe different models based on Recurrent Convolutional Neural Network with our model Recurrent Convolutional Neural Network on Sentences produces the best result which is on par with the state-of-the-art dialogue act recognition systems. It takes significantly less number of parameters to train it and it can be applied for speech recognition, text generation, machine translation and semantic analysis. Also, this is a comprehensive study of the methods that can be used for speech act recognition and extensive experimentation shows the results obtained for better insight into the usability of the model. Our future plans are to make it work on multi-label classification as well and combine character-aware neural network to the existing model.

# 7 References

[1] Chao Weng , Dong Yu, Shinji Watanabe, Biing-Hwang (Fred) Juang ,“Recurrent Deep Neural Networks For Robust Speech Recognition”

[2] Yoon Kim, “Convolutional Neural Networks for Sentence Classification”

[3] Guanghao Xu, Hyunjung Lee, Myoung-Wan Koo, Jungyun Seo, “Convolutional Neural Network using a Threshold Predictor for Multi-label Speech Act Classification”

[4] akashi Ushio, Hongjie Shi, Mitsuru Endo, Katsuyoshi Yamagami and Noriaki Horii, “Recurrent Convolutional Neural Network for structured speech act tagging”

[5] Yoon Kim, Yacine Jernite, David Sontag, Alexander M. Rush, “Character-Aware Neural Language Models”

[6] Zheqian Chen, Rongqin Yang, Zhou Zhao, Deng Cai, Xiaofei He, “Dialogue Act Recognition via CRF-A entive Structured Network”

[7] Ming Liang, Xiaolin Hu, “Recurrent Convolutional Neural Network for Object Recognition”