Comparison Among the Performance of Three Models in Intent Classification on the HarperValleyBank Dataset

Md. Al-Imran Abir
Department of Electrical and Electronic Engineering,
Bangladesh University of Engineering & Technology (BUET), Dhaka, Bangladesh

***Abstract*—User intent classaification is an important natural language understanding task. In this project, three models were used to predict caller intents using the transcripts provided in the HarperValleyBank dataset which is a spoken dialog corpus that contains audio data along with transcripts and annotations for speaker identity, caller intent, dialog actions, and emotional valence. Two models were built using logistic regression classifiers that were trained by BERT and DistilBERT embedding of text data. The other model was a convolutional neural network (CNN) based architecture. The DistilBERT based model performed best followed by CNN and BERT based models. The DistilBERT based model also outperforms other models reported in literature for the same dataset with an accuracy of 95.60%.**

***Index Terms*—BERT, CNN, DistilBERT, HarperValleyBank**

I. Introduction

Spoken Language Understanding (SLU) is the task of inferring the semantic meaning of spoken utterances. SLU is an essential component of voice assistants, social bots, and intelligent home devices [1], [2] which have to map speech signals to executable commands every day.

Natural Language Understanding (NLU) is a key part of the SLU pipeline that predicts the semantics (domain, intent and slots) from the utterance transcript. NLU is critical to the performance of goal-oriented spoken dialogue systems. One of the fundamental task in NLU is intent classification.

Intent classification focuses on the task of extracting intents from queries and conversations. The goal is to assign labels to text. It has broad applications including topic labeling, sentiment classification, and spam detection, etc. Intent classification is a classification problem that predicts the intent label $y^{(i)}$ from a given text sequence $x^{(i)} \in \mathbb{R}^n$.

In open-domain conversations, context information (one or a few previous utterances) is particularly important to language understanding [3] . The real spoken dialogue scenario always have multiple turns, and the number of back and forth between both sides increases as the complexity of the scenarios grows. The accurate understanding of next dialogue sentence often requires reasoning from its previous conversational history which is referred as context. Failing to consider the contextual information may result in incorrect interpretation of the user's intent.

Recurrent neural networks (RNN) that have a recurrent connections of hidden layer activities once dominated the field of NLU and thus intent classification([4]–[7]) as they can consider the contextual information to some extent. However, RNNs have some problems like inability to handle more than 10-20 time steps (vanishing gradient problem), condensing all the information in the input sentence into one hidden state vector, etc. Some of these problems were solved by the Long Short-Term Memory (LSTM) [8] which became popular for NLU tasks [9], [10].

After introduction of attention based transformer architecture [11], different transformer based models achieved state-of-the-art performance.

Lack of human-labeled data for NLU and other natural language processing (NLP) tasks results in poor generalization capability. To address the data sparsity challenge, a variety of techniques were proposed for training general purpose language representation models using an enormous amount of unannotated text, such as ELMo [12], Generative Pre-trained Transformer (GPT) [13], Bidirectional Encoder Representations from Transformers (BERT) [14], etc. Though GPT, BERT, etc have significantly improved the performance of many NLU tasks, they are computationally very expensive constraining their applications. There have been also some concerns about the environmental impacts of such models [15], [16]. So, there have been extensive work going on compressing these models while achieving more or less same performance compared to the main network. Some of such models are TinyBERT [17], MobileBERT [18], DistilBERT [19], DistilGPT-2[1], KnGPT2 [20], etc.

In this work, we developed three models for intent classification on the HarperValleyBank dataset [21]. This dataset contains audio data with transcripts where the dialogues simulate simple consumer banking interactions between users and agent. Each conversation between an agent and caller has a single intent/task and the models can predict that intent. Two of the models are based on BERT and DistilBERT models whereas the other one is a convolutional neural network (CNN) based model. We have found that all of these models outperform most of the existing models whereas the DistilBERT based model outperforms all previous reported models on the same dataset.

[1]https://huggingface.co/distilgpt2

II. Related Work

Deep learning based models have been extensively studied for NLU tasks. In [22], the author used a CNN model trained on top of word vectors for sentence-level classification tasks. In [23], the authors used character-level convolutional networks for text classification. RNN and LSTM models were used for utterance classification in [24]. In [25], hierarchical attention networks were used for document classification. All of these models were trained and tested on datasets other than the one we are using.

As HarperValleyBank is a relatively new dataset, there hasn't been much work on this dataset. We have found four prior works on this dataset related to intent classification. In [21], the authors used three models for intent classification. The first one is a bi-directional LSTM with connectionist temporal classification or CTC [26] loss function. The second one is Listen-Attend-Spell (LAS) network [27] where the listener network is composed of three stacked pyramid bi-directional LSTMs with 128 hidden dimensions whereas the speller network is an uni-directional LSTM with 256 hidden dimensions and a single-headed attention layer. The third one is a multi-task objective combining the two previous losses [28]. In [29], the authors used RNN transducer model for intent classification. In [30], they used BERT embeddings with an RNN transducer model for this task whereas in [31], the authors used ESPnet [32] for the same task.

III. Proposed Approach

In this project, three models were trained and evaluated using the dataset and their performances were compared. In one model, we used BERT embeddings to train a multiclass logistic regression (LR) model. In another similar model, we used DistilBERT embeddings instead of BERT to train a multiclass LR model. The third model is a convolutional neural network (CNN) based architecture. Brief description about these models are provided in the following subsections.

A. BERT

The model architecture of BERT [14] is a multi-layer bidirectional Transformer encoder based on the original Transformer model [11].The input representation is a concatenation of WordPiece embeddings [33], positional embeddings, and the segment embedding. Specially, for sentence classification and tagging tasks, the segment embedding has no discrimination. A special classification embedding ([CLS]) is inserted as the first token and a special token ([SEP]) is added as the final token. Given an input token sequence $\boldsymbol{x} = (x1, ..., xT)$, the output of BERT is $\boldsymbol{H} = (h1, ..., hT)$.

The BERT model is pre-trained with two strategies on large-scale unlabeled text, i.e., masked language model and next sentence prediction. The pre-trained BERT model provides a powerful context-dependent sentence representation and can be used for various target tasks, for example intent classification, similar to how it is used for other NLP tasks.

Two models of different size was pre-trained. One is the $\text{BERT}_{\text{BASE}}$ model with 12 layers (Transformer blocks), 768 hidden states, and 12 attention heads. The second one is called $\text{BERT}_{\text{LARGE}}$ which has 24 layers, 1024 hidden states and 16 attention heads. $\text{BERT}_{\text{BASE}}$ has total 110 million parameters whereas $\text{BERT}_{\text{LARGE}}$ has 340 million parameters. In this work, $\text{BERT}_{\text{BASE}}$ was used.

B. DistilBERT

The DistilBERT model, as evident from its name, is a distilled version of the BERT model [19]. This model uses knowledge distillation technique [34], [35], also known as teacher-student learning, to compress the original BERT model. In this technique, a compact model - the student - is trained to reproduce the behaviour of a larger model - the teacher - or an ensemble of models. The student is trained with a distillation loss over the soft target probabilities of the teacher:

$$L_{CE} = \sum_i t_i \times \log(s_i) \tag{1}$$

where t_i (resp. s_i) is a probability estimated by the teacher (resp. the student).

Also, a softmax temperature is used:

$$p_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \tag{2}$$

where T controls the smoothness of the output distribution and z_i is the model score for the class i. The same temperature T is applied to the student and the teacher at training time, while at inference, T is set to 1 to recover a standard softmax.

The final training objective is a linear combination of the distillation loss L_{CE} with the supervised training loss (the masked language modeling loss L_{mlm} [14]). Also, a cosine embedding loss (L_{cos}) is added which tends to align the directions of the student and the teacher hidden states vectors.

The student - DistilBERT - has the same general architecture as BERT. The token-type embeddings and the pooler are removed while the number of layers is reduced by a factor of 2.

DistilBERT has 40% less parameters than $\text{BERT}_{\text{base}}$-uncased, runs 60% faster while preserving over 95% of BERT's performances as measured on the GLUE language understanding benchmark [36].

C. Classification Using BERT and DistilBERT

The BERT and DistilBERT model output embeddings for each word uttered in a conversation along with a special classification token(`[CLS]` denoted by C) at the beginning of each sequence. The final hidden state corresponding to this token is used as the aggregate sequence representation for classification tasks. The final hidden vector of the special `[CLS]` token (C$\rightarrow \mathbb{R}^H$ where H is the number of hidden units) is then used to train a multiclass logistic regression model that tries to minimize the following loss function (Fig. 3):

$$J(\theta) = -\Big[\sum_{i=1}^{m}\sum_{k=1}^{K} 1\{y^{(i)} = k\} \log \frac{\exp(\theta^{(k)T} x^{(i)})}{\sum_{j=1}^{K} \exp(\theta^{(j)T} x^{(i)})}\Big] \tag{3}$$

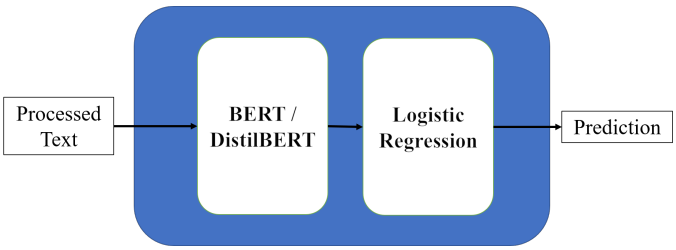

Fig. 1. Workflow for the BERT and DistilBERT based model.

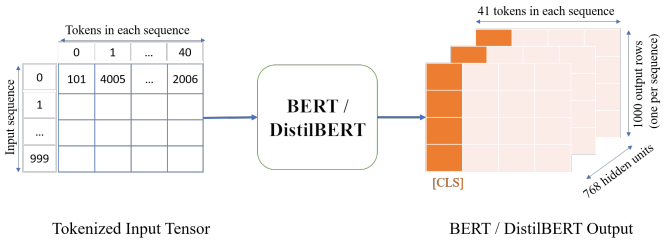

Fig. 2. Input and output of BERT and DistilBERT model.

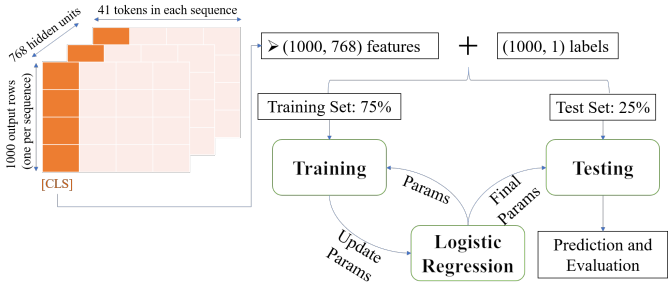

Fig. 3. Training and testing of the logistic regression model with BERT or DistilBERT output.

where $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\}$ are m labeled training examples with input features $x^{(i)} \in \mathbb{R}^n$ and labels $y^{(i)} \in \{1, 2, \ldots, K\}$ and K is the number of classes.

D. CNN based Model

The architecture of the CNN based network is summarized in Fig. 4 and Table I. The network contains six 1D convolutional and two fully-connected (FC) neural network layers. Each of the first four convolutional layers has 64 kernels of size 5. The last two convolutional layers each have 128 kernels of size 3. After two convolutional layers, a max pooling layer with a pool size of 2 is used. The FC layers have 100 neurons each. The output of the last FC layer is fed to a 8-way Softmax which produces a distribution over the 8 class labels.

IV. Experiments

A. Data

In this project, the HarperValleyBank corpus [21] was used for training and testing the models. The dataset is a free, public domain spoken dialog corpus, where the dialogs simulate simple consumer banking interactions between users and agents. There are 1,446 human-human conversations (23 hours of audio) between 59 unique speakers in the original dataset but in the GitHub repository[2] only 1000 were uploaded. There are transcripts provided for each utterance, in addition to annotations for speaker identity, caller intent, dialog actions, and emotional valence.

In this work, focus was to predict caller intent. This task attempts to predict a single intent that represents the customer's goal in the conversation. Each conversation is labelled with one of eight categories: order checks, check balance, replace card, reset password, get branch hours, pay bill, schedule appointment, or transfer money. The distribution of the intents is roughly balanced as shown in Fig. 5.

Each conversation was provided in a separate JSON file as an array of nested objects. Each utterance is given under the key `human_transcript` and the role of the speaker of each utterance is given under the key `speaker_role`. All the utterances of a single conversation were accumulated and with all the conversation a Pandas dataframe was created before further processing.

The intent label for each conversation was provided in other separate JSON file under the `task_type` key of the `tasks` key (nested objects). These JSON files contain metadata for respective conversation.

TABLE I: The CNN based model summary showing each layer's output shape and no of parameters

Type	Shape	Param #
InputLayer	[(None, 430)]	0
Embedding	(None, 430, 60)	39840
Conv1D	(None, 426, 64)	19264
Conv1D	(None, 422, 64)	20544
MaxPooling1D	(None, 211, 64)	0
Conv1D	(None, 207, 64)	20544
Conv1D	(None, 203, 64)	20544
MaxPooling1D	(None, 101, 64)	0
Conv1D	(None, 99, 128)	24704
Conv1D	(None, 97, 128)	49280
MaxPooling1D	(None, 48, 128)	0
Flatten	(None, 6144)	0
Dense	(None, 100)	614500
Dense	(None, 100)	10100
Dense	(None, 8)	808
Total trainable parameters: 820,128		

B. Evaluation Metrics

As the dataset is roughly balanced for caller intent labels (Fig. 5), all of the previous reported works had used only accuracy for evaluation. However, as the dataset is not fully

[2]https://github.com/cricketclub/gridspace-stanford-harper-valley

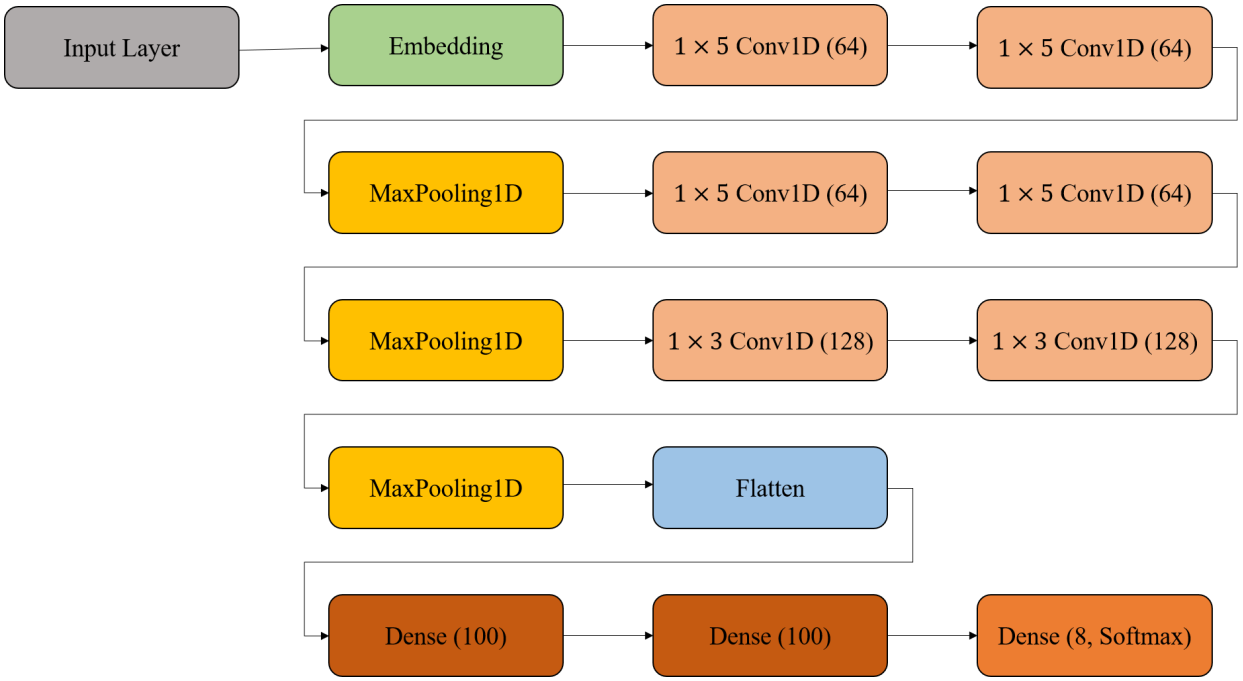

Fig. 4. Architecture of the CNN model.

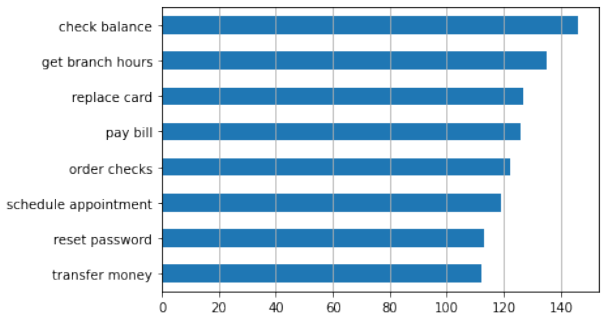

Fig. 5. Distribution of caller intent (shows the count for each intent).

balanced, we have also used the F1-score for evaluation. F1-score is defined as

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{4}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

where TP is no of true positives, FP is no of false positives and FN is no of false negatives.

C. Model Training

1) BERT and DistilBERT Based Model: Before passing the data through the BERT and the DistilBERT based models, the text data need to be tokenized. For tokenizing each conversation into a sequence, we used the `BertTokenizer` and the `DistilBertTokenizer` available in the HuggingFace Transformer library respectively. After tokenization, all sequences didn't have equal length. For processing by BERT and DistilBERT, all of them should be of equal length. So, they were made of equal lengths by padding zero to the right of each sequence. After zero-padding, each sequence had a length of 430. But the computing resources necessary to process data of this length weren't available. So, we inspected the dataset manually, and found that the first portion of each conversation was essential for intent recognition as the caller usually tells his intent at the beginning of the call. So, we kept only the first 41 tokens along the special `[CLS]` token and passed these 41 tokens through BERT and DistilBERT for processing. The input to the BERT and DistilBERT had a shape of 1000×41 where 1000 is the number of samples available in the dataset whereas the output of the BERT and the DistilBERT had a shape of $1000 \times 768 \times 41$ where 768 is the number of hidden states as shown in Fig.2. Of the 41 tokens, we used only the first special `[CLS]` token for classification. So, the feature size was 1000×768 (Fig. 3). We then randomly divided this 1000 samples into training and test set. In total, 750 samples were used for training. We didn't need to process the categorical labels as the Logistic Regression model of Scikit-learn[3] automatically handles that.

2) CNN based Model: In this case, we use the Keras `Tokenizer` class[4] for tokenizing the text sequences. Also, the categorical labels were handled using

[3]texthttps://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[4]https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer

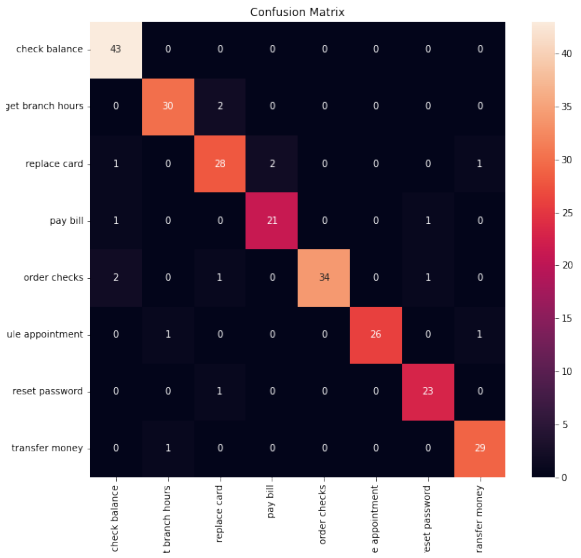

Fig. 6. Confusion matrix for the BERT model.

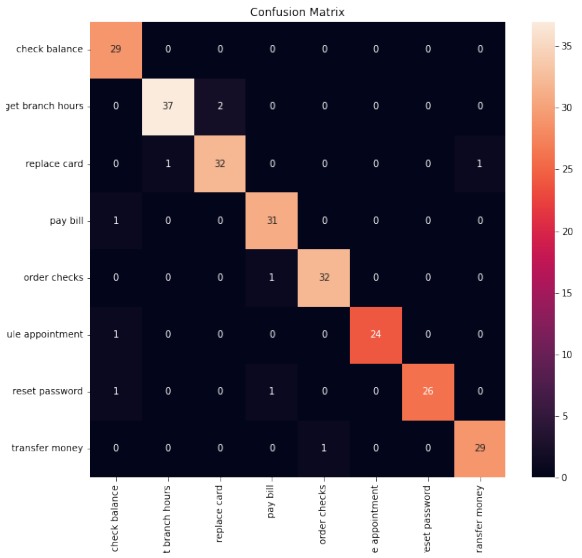

Fig. 7. Confusion matrix for the DistilBERT model.

the `to_categorical` class of Keras. Then the unequal sequences were made of equal length (430) by padding zero to the right of each sequence. Then we split all samples in training (68%), validation (15%), and test set (17%). Using the training set, we trained the model shown in Fig. 4. With the help of the validation set, we tuned the hyper parameters (convolutional kernel size and numbers, no of hidden states, no of neurons in fully-connected layers, use of pooling, batch size, no of epochs etc.) For all convolutional and FC layers, ReLU activation was used, except for the output layer where softmax activation was used. There were 820,128 trainable parameters in total in the model.

V. Results

For the BERT model, we got a training accuracy of 100% and test accuracy of 92.00%. As the model is roughly balanced, the F1-score has been also calculated and it was also 92.00%. The corresponding confusion matrix is shown in Fig.6.

For the DistilBERT model, we got a training accuracy of 99.20% and test accuracy of 95.60%. The F1-score was 94.80% in this case. The corresponding confusion matrix is shown in Fig. 7.

While training the CNN based model, we used "Early Stopping" of Keras and restored the best weights. Due to the early stopping, the training was done for 17 epochs and we got best parameters at 12th epoch. The training and validation losses are shown in Fig. 8 and the training and validation accuracy are shown in Fig. 9. The training, validation, and test accuracy were 98.97%, 94.12%, and 94.00%. The test F1-score was 94.12%. Due to the random nature of the models, all the models were run for five times and the performance metrics reported here are the median of those five values.

The test accuracy and F1-scores for these three models along with other reported works of intent prediction on this dataset is shown in Table II. As can be seen from the Table II, the DistilBERT models outperforms other models.

Also, among the three models inspected in this paper, the BERT based model performs worse than others which is unusual. Usually, BERT perform better than DistilBERT as shown in [19]. In this case BERT falls behind DistilBERT most probably due to over-fitting.

Another interesting fact is the performance of the CNN based model. Usually, for text data CNN is not supposed to perform in par with the models like BERT and DistilBERT. It is perhaps due to the fact that for intent classification from the dataset we are using, there is no need for global/long-range semantics. It is possible to detect the intent of a conversation by considering only few neighboring words as the dataset is an artificial one where conversations were done following some templates.

VI. Conclusion

In this work, we proposed three models for intent classification. One was based on the BERT model, another one on the DistilBERT model and the other one was a CNN based model. For the BERT and DistilBERT based models, we used logistic regression classifiers which were trained by the special classification token output from the BERT and the DistilBERT models. These models could predict intent of a conversation between a caller and bank agent with the accuracy of 92.00%, 94.00%, and 95.60% respectively. As the dataset was slightly unbalanced, we also reported the F1-scores of the models. The performance of these models on the HarperValleyBank dataset were better than all other previously reported works. In this work we got this high accuracy without any fine-tuning of the

TABLE II: Comparison among different model's performances.

Reference	Data type	Model	Accuracy (%)	F1 Score (%)
This work	Text only	CNN	94.00*	94.31*
		BERT	92.00*	92.00*
		DistilBERT	**95.60***	**94.80***
[21]	Text and audio both	LSTM (CTC loss)	45.47	N/A
		Listen-Attend-Spell	34.96	N/A
		LSTM(CTC) + Listen-Attend-Spell	42.28	N/A
[29]	Text only	RNN Transducer	76.97	N/A
	Text and Audio both		89.89	N/A
[30]	Text and audio both	BERT + RNN Transducer	94.00	N/A
[31]	Text and audio both	ESPnet-SLU	47.10	N/A

* Median value after 5 runs of the model

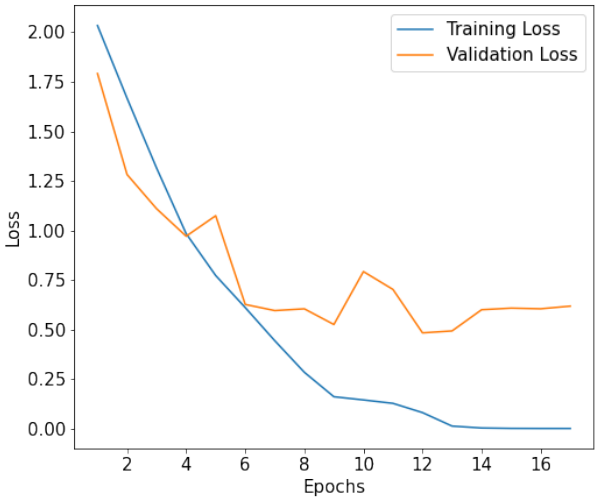

Fig. 8. Training and validation loss for the CNN model.

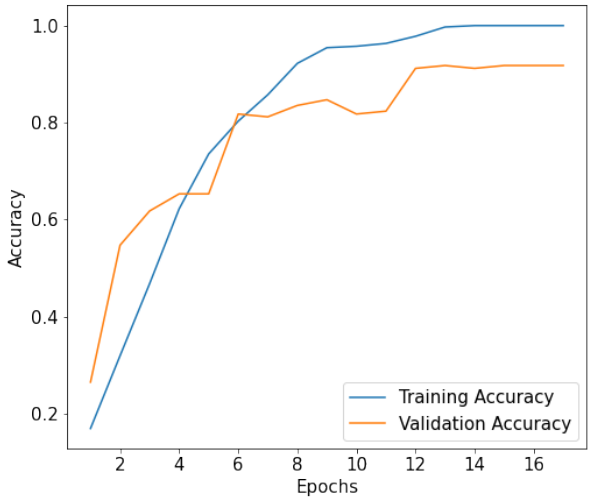

Fig. 9. Training and validation accuracy for the CNN model.

BERT and DistilBERT model. In future works, these model can be fine-tuned for even higher accuracy.

REFERENCES

[1] D. Yu, M. Cohn, Y. M. Yang, C.-Y. Chen, W. Wen, J. Zhang, M. Zhou, K. Jesse, A. Chau, A. Bhowmick *et al.*, "Gunrock: A social bot for complex and engaging long conversations," *arXiv preprint arXiv:1910.03042*, 2019.

[2] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril *et al.*, "Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces," *arXiv preprint arXiv:1805.10190*, 2018.

[3] C. Liu, P. Xu, and R. Sarikaya, "Deep contextual language understanding in spoken dialogue systems," in *Sixteenth annual conference of the international speech communication association*, 2015.

[4] Y. Shi, K. Yao, H. Chen, Y.-C. Pan, M.-Y. Hwang, and B. Peng, "Contextual spoken language understanding using recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5271–5275.

[5] M. Auli, M. Galley, C. Quirk, and G. Zweig, "Joint language and translation modeling with recurrent neural networks," in *Proc. of EMNLP*, 2013.

[6] T. Mikolov and G. Zweig, "Context dependent recurrent neural network language model," in *2012 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2012, pp. 234–239.

[7] K. Yao, G. Zweig, M.-Y. Hwang, Y. Shi, and D. Yu, "Recurrent neural networks for language understanding." in *Interspeech*, 2013, pp. 2524–2528.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] H. Sheil, O. Rana, and R. Reilly, "Predicting purchasing intent: Automatic feature learning using recurrent neural networks," *arXiv preprint arXiv:1807.08207*, 2018.

[10] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*. Ieee, 2013, pp. 6645–6649.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[12] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee,

and L. Zettlemoyer, "Deep contextualized word representations," 2018. [Online]. Available: https://arxiv.org/abs/1802.05365

[13] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding with unsupervised learning," 2018.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[15] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," *arXiv preprint arXiv:1906.02243*, 2019.

[16] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green AI," *CoRR*, vol. abs/1907.10597, 2019. [Online]. Available: http://arxiv.org/abs/1907.10597

[17] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "Tinybert: Distilling bert for natural language understanding," *arXiv preprint arXiv:1909.10351*, 2019.

[18] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, "Mobilebert: a compact task-agnostic bert for resource-limited devices," *arXiv preprint arXiv:2004.02984*, 2020.

[19] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[20] A. Edalati, M. Tahaei, A. Rashid, V. P. Nia, J. J. Clark, and M. Rezagholizadeh, "Kronecker decomposition for gpt compression," *arXiv preprint arXiv:2110.08152*, 2021.

[21] M. Wu, J. Nafziger, A. Scodary, and A. Maas, "Harpervalleybank: A domain-specific spoken dialog corpus," *arXiv preprint arXiv:2010.13929*, 2020.

[22] Y. Kim, "Convolutional neural networks for sentence classification," *CoRR*, vol. abs/1408.5882, 2014. [Online]. Available: http://arxiv.org/abs/1408.5882

[23] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *Advances in neural information processing systems*, vol. 28, 2015.

[24] S. Ravuri and A. Stolcke, "Recurrent neural network and lstm models for lexical utterance classification," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[25] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2016, pp. 1480–1489.

[26] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[27] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[28] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 4835–4839.

[29] S. Thomas, H.-K. J. Kuo, B. Kingsbury, and G. Saon, "Towards reducing the need for speech training data to build spoken language understanding systems," *arXiv preprint arXiv:2203.00006*, 2022.

[30] J. Ganhotra, S. Thomas, H.-K. J. Kuo, S. Joshi, G. Saon, Z. Tüske, and B. Kingsbury, "Integrating dialog history into end-to-end spoken language understanding systems," *arXiv preprint arXiv:2108.08405*, 2021.

[31] S. Arora, S. Dalmia, P. Denisov, X. Chang, Y. Ueda, Y. Peng, Y. Zhang, S. Kumar, K. Ganesan, B. Yan *et al.*, "Espnet-slu: Advancing spoken language understanding through espnet," *arXiv preprint arXiv:2111.14706*, 2021.

[32] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, "Espnet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, 2018.

[33] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[34] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 535–541.

[35] G. Hinton, O. Vinyals, J. Dean *et al.*, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, vol. 2, no. 7, 2015.

[36] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," *arXiv preprint arXiv:1804.07461*, 2018.