Deep Learning For Recognizing Human Confusion

Anushree Das
Department of Computer Science
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY 14586
ad1707@rit.edu

***Abstract*—To build a more human-like intelligent system, it needs to be able to predict the emotions of people it interacts with. Confusion is a complex emotion, which is less incomprehensible when compared with the more common emotions like sadness and anger. This project explores the method of down-streaming a pre-trained language representations model using the fine-tuning approach. The proposed model for this project is a fine-tuned BERT model trained on transcribed speech data which can detect confusion in humans.**

***Index Terms*—Natural Language Processing; Confusion Detection;**

I. Introduction

Emotions are expressed when interacting and socializing with other people and humans are capable of detecting those emotions just from the context of a sentence or phrase. Similarly, it is important for an intelligent system to be able to accurately predict the emotions of people it interacts with, in order to build a more human-like intelligent system. Natural language processing (NLP), which refers to the branch of artificial intelligence concerned with giving computers the ability to understand text and spoken words in much the same way human beings can. A deep learning model can be trained on transcribed speech words to detect the underlying emotion behind it.

There are many emotions recognition models already available which can detect the emotions like anger, sadness, joy, fear, disgust, etc. Confusion is a complex emotion, which is less incomprehensible when compared with the more common emotions like sadness and anger, therefore this project will be focusing on the task of recognizing confusion in humans. Recognizing confusion has become an essential factor in improving various kinds of tasks, for instance, analyzing the students' facial expression or their linguistics to detect confusion automatically, during lectures, can help educators effectively intervene and clear the doubts to help improve the students' academic performance, AIs like virtual assistants can be improved if it gains the capability to detect human confusion while it interacts with a human.

The meaning of confusion according to the Oxford dictionary is "a state of not being certain about what is happening, what you should do, what something means, etc." Confusion can stem from disagreements, unclear information, and pragmatically unexpected behaviors such as speaking about topics out-of-context [6]. When people are confused, they usually furrow, scrunch, or lower their eyebrows and purse or bite their lips [1] [2]. The occurrence of disfluencies such as repairs, filled pauses, or silent pauses are indicators of confusion in humans as well [6]. Presence of words like "don't understand" or "what?" is also an indicator of confusion [3].

There have been few studies on recognizing confusion in human, for example, Lago (2014) [1] and Borges (2019) [2] used features extracted from facial expressions to detect confusion. Atapattu (2019) [3] and Zeng (2017) [4] explored detecting confusion using linguistic features extracted from textual data. Research conducted by Kaushik (2012) [6] trained Random Forest model using features extracted from facial expressions and transcribed speech data, to detect confusion. Mince (2012) [7] extended Kaushik (2012) [6]'s work by adding one more mode of features (audio signals) and built learning models using Hidden Markov Models(HMM).

Due to the lack of enough data to train a fully-fledged neural network, this project explores the method of down-streaming a pre-trained language representations model using the fine-tuning approach. A BERT [8] model is fine-tuned using the transcribed speech data used by Mince (2012) [7].

In the following section, prior work is discussed in detail along with their results. Section III contains an overview of the dataset used, the Bert model, and its parameters. Section IV describes the evaluation method and shows the results for fine-tuning different types of Bert models. Section V discusses the observations from the results obtained from fine-tuning different types of Bert models and compares them. Section VI concludes the findings from this study and provides suggestions for future work.

II. Related Work

Borges (2019) [2] used Facial Action Coding System (FACS) to extract features from facial expressions. FACS is a comprehensive, anatomically based system that breaks down facial expressions into individual components of muscle movement, called Action Units (AUs). The goal of Borges (2019) [2] study was to build an LSTM neural network that uses the temporal context of facial expression to classify affective states like Confusion, Interest, and Boredom, and prove that a model trained on time series data of relevant Action Units performs better than a model which doesn't take the temporal context of facial features into account. Lago (2014) [1] came up with a computational model of emotion, based on facial expression

analysis, which can be used for inferring Confusion, Boredom, and Interest, by developing their own heuristics that associate AUs to emotion. Lago (2014) [1] 's model is based on changes in the facial expression of the user with respect to a baseline, which allows the model to adapt to different users, moods, and personal manners. Both Borges (2019) [2] and Lago (2014) [1] proved that temporal context is important for detecting affective states like confusion. Our study tries to take into account the temporal context of data, by learning context from an utterance and classifying it.

Atapattu (2019) [3] and Zeng (2017) [4] used the Stanford MOOC post dataset [5] to build models which could detect confusion in different domains. MOOC post dataset [5] comprises of student textual posts from courses and their corresponding confusion scores. Atapattu (2019) [3] trained random forest model on linguistic features to classify confusion into two categories: direct and indirect, separately for different domain data and achieved 83.1, 94.5, and 85.1 of F- measure for the domains: Humanities, Education, and Medicine respectively. Zeng (2017) [4] trained model using content-related (e.g. post length, readability score) and community-related (e.g. votes, reads) features and achieved over 80% of classification accuracy for individual domains (e.g. education). Both, Atapattu (2019) [3] and Zeng (2017) [4] learned from the features extracted from text, indicating the importance of text data features in detecting confusion. Our study learns the context from the text data to classify it into different levels of confusions.

Research conducted by Kaushik (2012) [6] and Mince (2012) [7] used different modes of features extracted from zoom meetings. More about the dataset is explained in the next section. The different types of features extracted from zoom meeting recordings for the study in Kaushik (2012) [6] were facial expressions and transcribed speech data. The result of the bimodal (transcribed speech data and facial expression) learning model was compared with the results of unimodal(speech and facial expression, separately) learning models to prove the efficiency of multimodal learning. Mince (2012) [7] extended Kaushik (2012) [6]'s work by experimenting with three modes of data (audio, video, and transcribed speech). They trained different types of models like Gaussian HMM, Gaussian Mixture model, models built using different types of transition matrices, unimodal (audio, video, and text separate) models, bimodal (pairs from audio, video, and text) models, and trimodal model. The results for all the models were compared, it was observed that the performance of the trimodal model was the best. The study concluded with the suggestion that a Recurrent Neural Network (RNN) can overcome the issues faced by a classifier built using HMM since the classification in RNN is based on a series of events, contrary to the HMM, in which a state is dependent only on the immediate past state. Both Kaushik (2012) [6] and Mince (2012) [7] proved that transcribed speech data is an important feature for detecting confusion. Mince (2012) [7] further proved that the temporal context of the data is important for recognizing confusion.

TABLE I
SAMPLE FROM PROCESSED DATASET

Utterance	Label
That doesn't work	0
Why is that	2
Right because that would obviously make it way easier	0
Um Should we like total it up	1
Okay let's do it	0
But you don't know which ones they are	1
Uh No I don't have that one	3
I don't know how you got there but I believe you	3
Mhm No , it kind of makes sense	1
Um I mean I can do any time this week	2

III. METHODS

A. Dataset

This project will use the dataset created in a prior study [7]. The data was collected through Zoom meetings, in which the participants were given three tasks that elicit confusion. Then the participants were asked to label their video recording into 4 classes: Not Confused-0, Slightly Confused-1, Very Confused-2, and Extremely Confused-3. Three separate types of features were extracted from the data: facial expression (AUs), voice features (pitch, intensity, speech rate, and MFCCs per frame), and transcribed text. The output label is stored as time series data, i.e., the output label for each second of the conversation between the participants recorded is provided.

A BERT model takes a sequence of tokens, for example, a sentence, as its input, therefore, to fine-tune the BERT model, only the transcribed speech dataset is used which is split into utterances and phrases. The label for each utterance or phrase, is set to the highest label between the start and end time of the utterance or phrase, for example, an utterance "Here's what I don't understand" has labels 2 and 3 between the utterance start time and end time, so the output label for this utterance is 3 (Extremely Confused). A sample of the processed dataset is shown in table I.

The transcribed speech dataset was further divided into training, validation, and test dataset with the ratio of 70:20:10. Figure 1 shows that the class distribution for utterances is highly imbalanced. There were around 800 samples with the class label Slightly Confused, but only around 400 samples for Extremely Confused. To address this imbalance, some of the samples with the class label Slightly Confused were dropped to have a total of around 600 samples in the dataset to make it slightly balanced, before fine-tuning the BERT model.

B. Proposed Model

The proposed model for this project is a Neural Network trained on transcribed speech data, which will be able to recognize confusion in a dialogue. Since it will be difficult to train a full-fledged Neural Network from scratch, we will be fine-tuning a pre-trained BERT model.

C. BERT models

BERT, which stands for Bidirectional Encoder Representations from Transformers [8], is a transformer-based machine

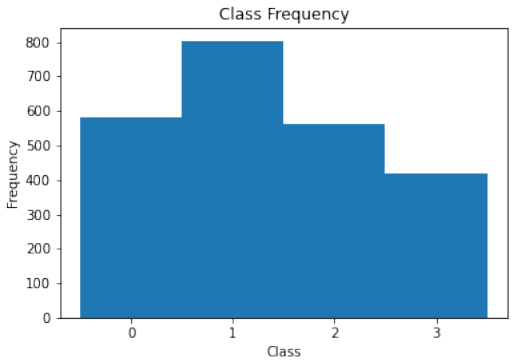

Fig. 1. Class Distribution, where Class 0, 1, 2, and 3 represent Not Confused, Slightly Confused, Very Confused, and Extremely Confused

learning technique for natural language processing (NLP) pre-training. Pre-training language representations is a process of training a general-purpose language understanding model on a large corpus (like Wikipedia), and then using it to downstream NLP tasks, like question answering task. The pre-trained BERT model can be fine-tuned with just one additional output layer to create new models for a wide variety of language tasks, such as question answering and language inference, without substantial task-specific architecture modifications. BERT was built upon other work in pre-training contextual representations — including Semi-supervised Sequence Learning, Generative Pre-Training, ELMo, and ULMFit — but crucially these models are all unidirectional or shallowly bidirectional. As opposed to directional models, which read the input sequentially (left-to-right or right-to-left), the BERT's transformer encoder reads the entire sequence of tokens at once. This characteristic allows the model to learn the context of a tokens based on all of its surroundings (left and right of the tokens). One reason why, BERT is able to model many downstream tasks, is because of the self-attention mechanism in the transformers[9]. A transformer is an attention mechanism that learns contextual relations between words (or sub-words) in a text and it includes two separate mechanisms - an encoder that reads the text input and a decoder that produces a prediction for the task.

During pre-training, the model is trained on unlabeled data over two different pre-training tasks, namely, Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) [8]. In order to train a deep bidirectional representation, the model is trained for MLM, a task in which some percentage of the input tokens are masked at random, and the model tries to predict those masked tokens. In order to learn relationships between sentences, the model is trained on the NSP task, which is, given two sentences A and B, predict whether B is the actual next sentence that comes after A, or just a random sentence.

The input to a BERT model is a sequence of tokens (figure 2), where tokens are simply words (or part of words, punctuation symbols, etc.) of the sentences. The first token of

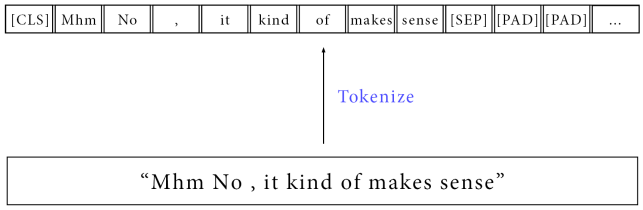

Fig. 2. An input sentence is converted to a sequence of tokens using a tokenizer method provided by the transformers library in python.

every sequence is always a special classification token ([CLS]). These tokens are converted into numbers, to be able to build a tensor out of them and are fed to the model. The architecture of a BERT-base model with 12 encoder layers is shown in figure 3. The embedding layer takes the sequence of tokens and encodes them into three sequences of the same length, which are added together and used as input to the self-attention layers (encoder layers). The pooler layer takes the final hidden state (output of the encoder) corresponding to [CLS] token and returns a fixed-dimensional pooled representation of the input sequence as output, which is further used as the aggregate sequence representation for classification tasks.

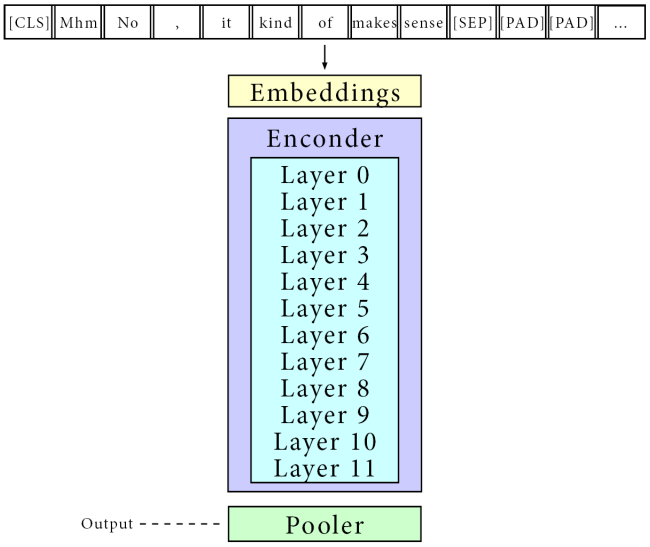

Fig. 3. Architecture of a BERT-base model with 12 encoder layers

There are many pre-trained models available for fine-tuning, for example, BERT [8], DistilBERT [10], ALBERT [11], TOD-BERT [12], etc. The original pre-trained BERT base model is a general-purpose language understanding model which is trained on a large corpus, whereas the dataset used for this project consists of dialogues. There was a possibility that a pre-trained model trained on dialogues might perform better, therefore, for this project, two base models, BERT base (uncased) and TOD-BERT, were fine-tuned and their performances were compared. TOD-BERT [12] is a pre-trained model which was trained on nine different dialogue-

TABLE II
ACCURACY FOR FINE-TUNED BERT MODELS ON TEST DATASET

Model	Accuracy (in %)
BERT-base-freeze12 (freeze 12 layers)	30
BERT-base-freeze6 (freeze first 6 layers)	80
BERT-base-2 (2 hidden layers)	76
BERT-base-6 (6 hidden layers)	80
BERT-base (12 hidden layers)	79
BERT-base-14 (14 hidden layers)	80
BERT-base-18 (18 hidden layers)	78
TOD-BERT-2 (2 hidden layers)	76
TOD-BERT-6 (6 hidden layers)	79
TOD-BERT (12 hidden layers)	78
TOD-BERT-14 (14 hidden layers)	78
TOD-BERT-18 (18 hidden layers)	78

based datasets like MetaLWOZ, Taskmaster, etc.

As mentioned before, the output of the final transformer layer is used as the features of the sequence to feed a classifier. The transformers library has the BertForSequenceClassification class which is designed for classification tasks. However, after testing the performance of that classifier, the resulting validation accuracy achieved while training turned out to be quite low since it consists of only one linear layer which takes an input of size 768 and classifies into 4 labels. Therefore, a single-hidden-layer (hidden layer size: 50 nodes) feed-forward neural network is used as the classifier for this project, which improved the validation accuracy while training from 37% to 80%.

The transcribed speech dataset was split into utterances and phrases, to fine-tune the BERT model. After experimenting, fine-tuning BERT model on utterances proved to be more effective than doing so on phrases. The speculated reason behind it is that since the BERT model consists of transformers that learn contextual relations between words, the more number of words helps gain better context. Another set of experiments conducted included comparing the performance of the two BERT models, BERT-base and TOD-BERT with different number of hidden layers as well as BERT-base model with some number of layers frozen (layer weights do not change while training). Different types of architectures of the BERT-base models are explained in figure 4.

IV. RESULTS

Various models were fine-tuned and their performance was compared to obtain the most suitable BERT model for this project. Results of BERT-base with all and some layers frozen, original BERT-base, original TOD-BERT, BERT-base Model with different number of hidden layers, TOD-BERT Model with different number of hidden layers is shown in table II. The different number of hidden layers tried during the experiments is 2, 6, 12 (original), 14 and 18. The number of layers more than 18 was not tried out because fine-tuning with a greater number of hidden layers will take a greater amount of time.

If two models have equally good accuracy, it would not matter from which model the predictions come. Therefore, to prove that one model is better than another one, statistical significance tests [13] can be carried out on the results of the two models. Statistical significance tests [13] give us a way of quantifying the probability that the difference between two systems is due to luck. It makes sure that the difference between the two algorithms, as observed in an individual comparison, is not coincidental. After referring to the decision tree for statistical significance test selection provided by Dror (2018) [13], the most suitable evaluation method for comparing two models for this project is the sign test. The test statistic is the number of examples for which algorithm A is better than algorithm B, and the null hypothesis states that given a new pair of measurements, for example, evaluations (a_i , b_i) of the two algorithms on a new test example, then a_i and b_i are equally likely to be larger than the other [13]. The assumption of this test is that the data samples are i.i.d, the differences come from a continuous distribution (not necessarily normal) and that the values are ordered. To get a list of evaluations for a model, accuracy scores for 20 samples from the test dataset were calculated. The list of differences in accuracy scores for a pair on model was plotted to confirm that it came from a continuous distribution, before calculating the p-value from them. For the given accuracy scores of two models (a_i , b_i), where $0 \leq i < 20$, the p-value for sign test depends on the count of instances when $a_i > b_i$, instances when $a_i = b_i$ and instances when $a_i < b_i$. P-value is a floating number that ranges between 0 and 1. The significance level, also denoted as alpha or α , selected as 0.05 for this project, is the probability of rejecting the null hypothesis when it is true. When model A is compared with model B, the null hypothesis is accepted if p-value is less than or equal to 0.05, stating model A is better than model B. The the null hypothesis is rejected if p-value is greater than 0.05, stating model A is not better than model B.

V. DISCUSSION

BERT-base model with different number of hidden layers were compared and the p-value for sign test for each pair of these models in shown in table III. If p-value is less than 0.05, it means model A is better than model B, and vice versa. From the results in table III, we can assume all BERT-base models are better than BERT-base-2, while BERT-base-6 is better than the rest of the BERT-base models. When the p-values for both conditions: model A is better than model B and model B is better than model A, are greater than 0.05, it means both model's performances are similar, and based on this, we can assume that BERT-base and BERT-base-14 are similar, since p-value for hypothesis BERT-base is better than BERT-base-14 is 0.9164 and p-value for hypothesis BERT-base-14 is better than BERT-base is 0.1796. Both BERT-base and BERT-base-14 are better than BERT-base-18. The best model, after comparing the p-values shown in table III, appears to be BERT-base-6.

In the next experiment, TOD-BERT model with different number of hidden layers were compared (table IV), resulting with TOD-BERT with 6 layers to be the best model. Similar to BERT-base, TOD-BERT with 2 encoder layers is the least efficient of all the TOD-BERT models compared. Hence, it

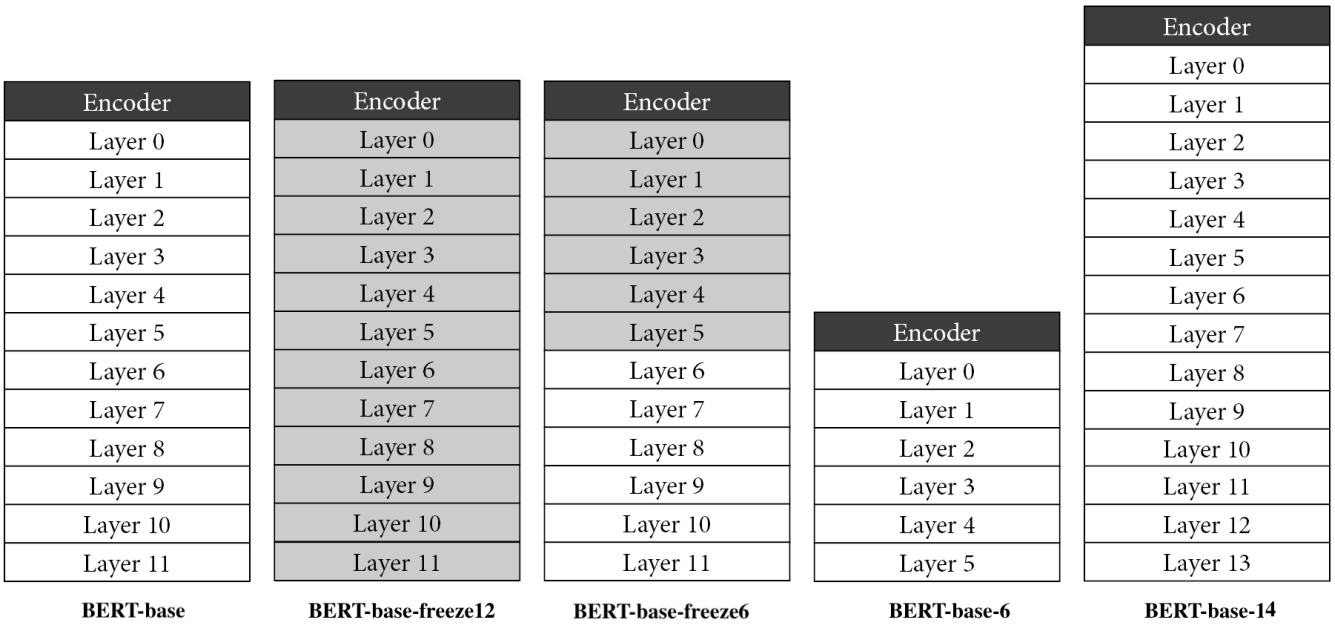

Fig. 4. The grey layers indicate that the layers have constant weights, i.e., the layers are frozen. 1. The BERT-base is the original BERT model with 12 layers in the encoder, 2. The BERT-base-freeze12 is the original BERT model whose encoder weights are not trained while fine-tuning, i.e., we freeze all the layers of encoder, 3. BERT-base-freeze6 is the original BERT model in which we freeze only the first 6 layers of the encoder, 4. BERT-base-6 is a BERT model with reduced number of encoder layers, 5. BERT-base-14 is a BERT model with increased number of encoder layers.

TABLE III
BERT-BASE MODEL WITH DIFFERENT NUMBER OF HIDDEN LAYERS

P-value	BERT-base-2	BERT-base-6	BERT-base	BERT-base-14	BERT-base-18
BERT-base-2 (2 hidden layers)	-	1.0	1.0	1.0	1.0
BERT-base-6 (6 hidden layers)	0.0000009	-	0.00003	0.0317	0.0000009
BERT-base (12 hidden layers)	0.0000009	0.9999	-	0.9164	0.0.0154
BERT-base-14 (14 hidden layers)	0.0000009	0.9903	0.1796	-	0.0.0154
BERT-base-18 (8 hidden layers)	0.0000009	1.0	0.9962	0.9962	-

can be said that any BERT model with 2 encoder layers is not as good at down-streaming a task as a BERT model with 6 or more layers. While, the p-value for comparing TOD-BERT with TOD-BERT-14 is 0.08 which is close to 0.05, we are going to assume that TOD-BERT is slightly better than TOD-BERT-14. Both TOD-BERT and TOD-BERT-14 are better than TOD-BERT-18.

When the best of the BERT-base and TOD-BERT model are compared (table V), the BERT-base model with 6 layers performs better than the TOD-BERT model with 6 layers.

Similarly, when BERT-base-6 was compared with BERT-base-freeze12 (all encoder layers are frozen) (table VI), p-value was 0.0000009, showing that BERT-base-6 is better than BERT-base-freeze12. Since, the p-values obtained from sign test for both hypothesis: BERT-base-6 is better than BERT-base-freeze6 and BERT-base-freeze6 is better than BERT-base-6, are greater than 0.05 (table VII), therefore, both, BERT-base-freeze6 and BERT-base-6, appear to be similar models. Since, BERT-base-6 has less number of layers in the encoder, it will be computationally less time consuming. Therefore, the most efficient model for the project is BERT-base-6.

VI. CONCLUSION

From the experiments performed above, we can conclude that the number of layers in the encoder of the BERT model highly affect the performance of the model. The most suitable

TABLE IV
TOD-BERT MODEL WITH DIFFERENT NUMBER OF HIDDEN LAYERS

P-value	TOD-BERT-2	TOD-BERT-6	TOD-BERT	TOD-BERT-14	TOD-BERT-18
TOD-BERT-2 (6 hidden layers)	-	1.0	0.9999	1.0	1.0
TOD-BERT-6 (6 hidden layers)	0.0000009	-	0.0154	0.0006	0.0717
TOD-BERT (12 hidden layers)	0.00002	0.9962	-	0.0835	0.3238
TOD-BERT-14 (14 hidden layers)	0.0000009	0.9999	0.9682	-	0.9518
TOD-BERT-18 (6 hidden layers)	0.0000009	0.9754	0.8203	0.1189	-

TABLE V
BERT-BASE VS TOD-BERT

P-value	BERT-base-6	TOD-BERT-6
BERT-base-6 (6 hidden layers)	-	0.000001
TOD-BERT-6 (6 hidden layers)	1.0	-

TABLE VI
BERT-BASE-6 VS BERT-BASE-FREEZE12

P-value	BERT-base-6	BERT-base-freeze12
BERT-base-6 (6 hidden layers)	-	0.0000009
BERT-base-freeze12 (freeze all 12 layers)	1.0	-

TABLE VII
BERT-BASE-6 VS BERT-BASE-FREEZE6

P-value	BERT-base-6	BERT-base-freeze6
BERT-base-6 (6 hidden layers)	-	0.7596
BERT-base-freeze6 (freeze first 6 layers)	0.4072	-

model for this project is the BERT-base model 6 number of layer in the encoder with an accuracy of 80% on the test dataset. Since, the accuracy scores for all the models were similar, the statistical significance test was useful in comparing the models. It was speculated in Section III that there is a possibility that a pre-trained model trained on dialogues might perform better, but it was not the case for this project, since BERT-base-6 model performed better than TOD-BERT-6.

For future work, it is suggested to explore affect of changing other parameters of the BERT model like the number of attention heads and the pooler function (tanh by default). It is also recommended to perform a other statistical significance test than sign test, since it only checks if algorithm A is better than B and ignores the extent of the difference.

REFERENCES

[1] P. Lago and C. Jimenez Guarin, *An Affective Inference Model based on Facial Expression Analysis*, in IEEE Latin America Transactions, vol. 12, no. 3, pp. 423-429, May 2014, doi: 10.1109/TLA.2014.6827868.

[2] N. Borges, L. Lindblom, B. Clarke, A. Gander and R. Lowe, *Classifying Confusion: Autodetection of Communicative Misunderstandings using Facial Action Units*, 2019 8th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW), 2019, pp. 401-406, doi: 10.1109/ACIIW.2019.8925037.

[3] T. Atapattu, K. Falkner, M. Thilakaratne, L. Sivaneasharajah, and R. Jayashanka, *An identification of learners' confusion through language and discourse analysis*, Language Processing (ICSLP), 2004, https://arxiv.org/abs/1903.03286, 2019.

[4] Z. Zeng, S. Chaturvedi, and S. Bhat, *Learner Affect Through the Looking Glass: Characterization and Detection of Confusion in Online Courses*, in Proceedings of the 10th International Conference on Educational Data Mining. 2017.

[5] A. Agrawal and A. Paepcke, *The Stanford MOOCPosts Data Set.*, https://datastage.stanford.edu/StanfordMoocPosts/, 2014.

[6] N. Kaushik, R. J. Bailey, A. G. Ororbia, and C. O. Alm, *Elicitation of confusion in online conversational tasks*, in Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czech Republic, 2021.

[7] C. Mince, S. Rhomberg, R. J. Bailey, A. G. Ororbia, and C. O. Alm, *Developing a Confusion-Detecting Intelligent Cognitive Agent*

[8] J. Devlin, M. Chang, K. Lee and K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones,A. Gomez,L. Kaiser and I. Polosukhin, *Attention Is All You Need* 2017 31st Conference on Neural Information Processing Systems (NIPS), 2017.

[10] V. Sanh, L. Debut, J. Chaumond and T. Wolf, *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*, 5th Edition Co-located with NeurIPS'19, 2019

[11] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma and R. Soricut, *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*, International Conference on Learning Representations (ICLR), 2020

[12] C. Wu, S. Hoi, R. Socher and C. Xiong, *TOD-BERT: Pre-trained Natural Language Understanding for Task-Oriented Dialogue*, Empirical Methods in Natural Language Processing (EMNLP), 2020

[13] R. Dror, G. Baumer, S. Shlomov and R. Reichart, *The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing*, 56th Annual Meeting of the Association for Computational Linguistics, 2018