Multimodal Dialogue Act Classification

Geet Shingi
6015146447
shingi@usc.edu

Samhitha Gundam
1866055918
gundam@usc.edu

Swapnil Arya
6456245219
swapnilg@usc.edu

ABSTRACT

A key step in Natural Language Understanding (NLU) for any conversational agent is classifying the general intent of the user utterance in a conversation, also known as Dialogue Act Classification (DAC). The majority of the existing work on this has only been done using lexical data, and through this project, we aim to leverage both the lexical and acoustic data to classify the dialogue acts. We expect that the acoustic features would provide additional features that can be leveraged for better performance in classification.

KEYWORDS

Dialogue Act Classification, Multimodal data, SWDA

1 INTRODUCTION

A Dialog act (DA) is an utterance in the context of a conversational dialog that serves a function in the dialog. Examples of dialogue acts can be questions, statements, backchannel, request, etc. A DA represents the meaning of an utterance at the level of illocutionary force, and hence, constitutes the basic unit of linguistic communication [12]. Dialog act essentially states the speaker's intent rather than the words' literary meaning.

Dialogue act classification (DAC) is the task of identifying the intended meaning behind a speaker's utterance in a conversation. The goal is to automatically classify the type of dialogue act a speaker performs, such as making a statement, asking a question, giving a command, expressing agreement or disagreement, providing an explanation, expressing gratitude, and so on. Dialogue act classification is critical for conversational agents and chatbots that interact with users via natural language. DAC helps these systems understand the intent of the user's input and generate appropriate responses.

Current approaches in DAC treat this as a text classification problem where speaker conversations are transcribed, and the lexical data is used to classify its dialog act. In this project, we leveraged both the lexical and acoustic data to classify the dialogue acts. We expected that the acoustic features would provide additional features and information about the text that can be leveraged for better performance in classification. Acoustic features refer to properties of the sound signal, such as pitch, duration, energy, spectral characteristics, etc., that can be extracted from the speech signal and used as input to machine learning algorithms for dialog act classification. For example, the pitch is a measure of the fundamental frequency of the voice. It can be used to distinguish between different types of utterances, such as questions (which typically have a rising pitch at the end) and statements (which typically have a falling or neutral pitch). Other acoustic features, such as energy and spectral characteristics, can be used to distinguish between different emotions in the speaker, which can be useful in classifying dialog acts such as expressing an opinion or expressing agreement or disagreement. Overall, acoustic features can provide valuable information for dialog act classification, particularly when the text transcription may be incomplete or inaccurate. Hence, we expected that acoustic signals would provide additional features that can be leveraged for better performance in classification but we found the case otherwise.

As part of this project, we implemented unimodal models for DAC employing text and speech modalities and built multimodal models to provide more information and signals for Dialogue Act Classification. We also perform a comparative study between multiple models to compare their efficacy.

2 RELATED WORK

Prior work on Dialogue act classification focuses on using lexical information for labeling the dialog acts. There are two major categories of methods for Dialogue Act Classification (DAC): one type considers it as a text classification task, where each statement is handled individually, and the other treats it as a sequence labeling problem[3, 6]. However, for this particular project, we will only be focusing on the text classification approach.

Recently various deep-learning approaches have shown promising results in DAC as a text classification problem. Previous implementations have built vector representations for each utterance and used deep neural networks like RNN, CNN or LSTM for classifying text [5, 7]. Some more implementations included hierarchical RNNs and CNN models[8, 9]. Later these works were extended using attention-based encoder-decoder architectures and transformer [10, 13]

Additionally, transfer learning techniques have been extensively used for DA classification, wherein human-human conversations are used to train models that have their uses in human-machine conversations [1]. These models have an accuracy of 75.34% on the Switchboard dataset. [4] proposed to integrate turn changes in conversations among speakers when performing DA classification. The aim was to incorporate the speaker-turns into encoding an utterance. For instance, in a dyadic conversation, given an utterance with dialogue act "Question" from speaker A, if the following utterance is from speaker B, then the corresponding act is likely to be "Answer"; however, if there is no change in speakers, then the following act is less likely to be "Answer".

In previous works, these models have explored different features, including lexical and syntactic features, prosodic cues, speaker interactions, and context information. In particular, some works have evaluated the use of context information, and the speaker turns for this task using various deep learning frameworks and transformers[1, 4, 9, 10].

In this current work, we plan to leverage these above models to work on textual data but also use acoustic feature data and build multimodal models for the same. There have also been models using various multimodal settings previously. [2] has used haptic actions along with text for dialog act classification. Additionally,

Table 1: SWDA Data examples

Utterance	DA Label	DA
Me, I am the legal department	sd	Statement-non-opinion
Uh-huh	b	Backchannel
I think it's great	sv	Statement-opinion
That's exactly it	aa	Agree/Accept
But, uh, yeah	%	Uninterpretable
Well, how old are you	qw	Wh-Question

[11] has used Non-verbal features (change of tone, facial expressions etc.) can provide cues to identify DAs, stressing the benefit of incorporating multi-modal inputs in the task.

3 METHOD

3.1 Data

For the purpose of this project, we are using the Switchboard Dialog Act (SWDA) Corpus. It is a collection of 1,155 five-minute telephone conversations between two participants, annotated with speech act tags. In these conversations, callers question receivers on provided topics, such as child care, recycling, and news media. 440 speakers participated in these 1,155 conversations, producing 199740 utterances labeled into 42 dialog acts. Some labels with dialogue acts can be seen in Table 1. The mean and max utterance length in the data is 9.62 and 132, respectively, with the mean and max dialogue length 172.94 and 457, respectively.

The 42-label SWDA corpus is split (stratified on labels) into 90:10 for training and testing. The conversations in SWDA are transcribed to provide lexical data, and the audio data needs to be preprocessed to generate acoustic features. For lexical data, we have used cleaned data from Nathan Duran's GitHub repository[1] to get us started.

Also, Fig. 1 shows that the data is heavily class-imbalanced as more than 90% of the data comprises of 9 labels statement-non-opinion, backchannel, statement-opinion, uninterpretable, agree/accept, appreciation, yes-no question, conversation-closing, wh-question. The figure does not contain all the labels as the rest of the labels have frequencies less than 1%.

3.2 Textual data

We tested a wide range of data pre-processing techniques, embeddings, and models for textual data as a comparative study. We also built a context-aware model only on textual data using RoBERTa/BERT CLS token features.

3.2.1 Data pre-processing. For data pre-processing, we experiment with combinations of lowercasing the text, non-alphabet character elimination, and contraction expansions. We skipped the stopwords removal process as stopwords may provide some crucial information about the dialogue act.

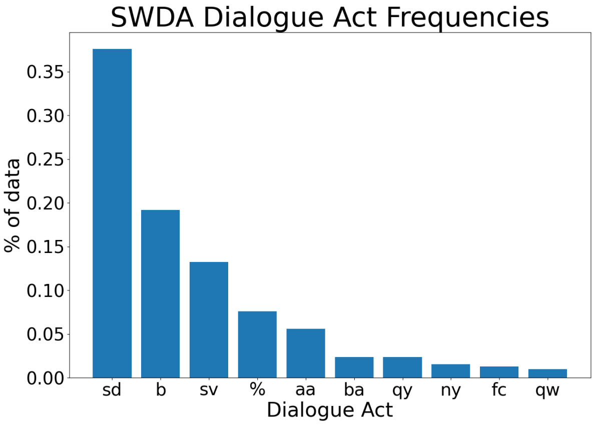

Figure 1: SWDA Label Frequencies

3.2.2 Model Implementation. We use a combination of different embeddings/features with multiple simple models, deep learning models, and transformer-based models. Due to class imbalance, all the models use weighted loss (for weighted weights). They are:

- Simple models: SVM, Logistic Regression, XGBoost, Perceptron with TF-IDF Vectorizer.
- Deep learning model: BiLSTM (1 layer and 2-layer with 50 hidden states) with Glove Twitter 100d and Glove Wikipedia 100d embeddings. In the case of 2-layer BiLSTM, a dropout of 0.2 was set.
- Transformers: BERT and RoBERTa CLS token features with simple models and 1-layer/2-layer/3-layer linear neural networks. We even tried using bert-base-cased with only expanding contractions.

We only used lowercasing and contraction expansion data preprocessing techniques for deep-learning and RoBERTa features models because that was the best performing.

3.2.3 Context-Aware Model. The architecture of context-aware model is show in Fig. 2. For the context-aware model we use previous 3 utterances as context along with target utterance. RoBERTa/BERT act as feature encoder and each utterance is passed through RoBERTa/BERT model to generate CLS token features which are passed to BiGRU model sequentially in order of context followed by a 2 fully-connected layers. The context-aware model performed the best across all textual, audio, and multimodal models. In this, we also encode speaker ID uisng a binary value 1 or 0 as only two speakers are present. The speaker ID is concatenated with the extracted features from BERT/RoBERTa encoder.

3.3 Audio data

3.3.1 Time Alignment and Label Generation. In the case of audio data, with the timestamped transcriptions, we aimed to extract speech features and labeled dialogue acts corresponding to certain segments of the transcript; however, this is not exactly straightforward as an utterance in transcription is split into multiple utterances in a dialog act, and also the text does not exactly match in both the files as can be seen in Table. 2. For DA classification with an audio

[1] https://github.com/NathanDuran/Switchboard-Corpus/tree/master/swda_data

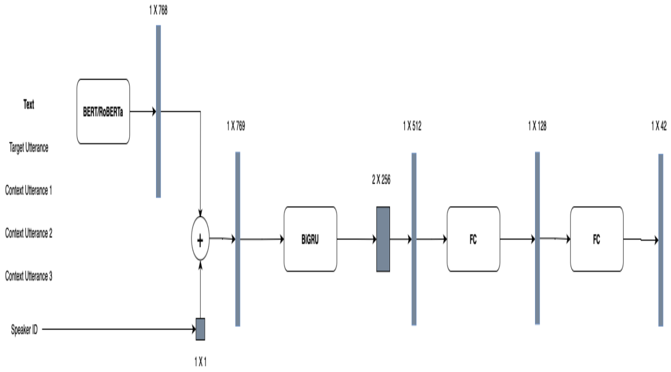

Figure 2: Context-Aware Model Architecture

or multimodal model, we will require audio files to have a labeled dialogue act. As a workaround, we use the word-level transcriptions to get granular level timestamps of spoken words by using heuristics such as edit distance and later merging them to form the spoken utterances.

The time alignment was an intensive task, not without its shortcomings. Several cases had to be handled individually depending on the significance of the token and the occurrence of the token in the utterance or the transcription. Utterances that were present in the dataset but not in the word-level transcriptions were ignored since these do not contain any timestamps. Both the transcripts and the utterance contained numerous filler words such as uhh, umm, umhum, uhhu, which could hint at the DA classification. Still, the filler words were not uniform between the utterance and transcriptions. We tried to unify all these words into a single word to have uniformity and to be more precise in timestamp alignments.

Lastly, in contrast to data-processing done for textual data, for pre-processing the transcript files for time alignment and syncing of the labels, we removed all non-spoken text tokens such as [SILENCE], [LAUGHTER], [NOISE], [VOCALIZED-NOISE] to have more accurate start and stop times for utterances.

3.3.2 Audio Feature Extraction. We extracted the audio for utterances in dialog acts using the above-aligned timestamps. We use features generated in 4 different methods and use them further to train the data for classification as shown in Fig 3. We explore all these methods below

- Audio embeddings using neural network - We already have existing features for pitch and fbank that we leveraged after aligning them to the dialog acts. These features were extracted using the method described in this paper [14]. We further applied different methods of pooling on these features - mean, max, 90 percentile and 95 percentile to obtain different results on this data
- Audio embeddings using VGGish - VGGish is a pre-trained convolutional neural network from Google whose architecture was inspired by the VGG networks used primarily for image classification. We used the VGGish network to generate audio embeddings. Before feeding the audio files to the model, there was a need to perform pre-processing tasks, such as Channel Splitting, since the Switchboard dataset contains two channels, A and B. Next step was to split each audio file into corresponding segments as per the time alignments extracted. The split audio files, in .wav format, were then provided as an input to the VGGish network. We used a pre-trained pipeline that extracts features of a given audio file using a VGGish model implemented in Pytorch[2]. This is a supervised model pre-trained with AudioSet, which contains over 2 million sound clips. The pipeline employs two operators, an audio decoder and an audio embedding operator that uses VGGish network to get audio embeddings. The audio decoder loads the audio files as a list of audio frames in ndarray, which are then combined to represent an audio clip in fixed length. Each array represents features of a non-overlapping clip with a fixed length of 0.96s and each clip is composed of 64 mel bands and 96 frames[3]. Finally the VGGish operator generates audio embeddings for each audio clip.
- Audio embeddings using MFCC generated features - In this method, we used librosa to load the data and generate features using MFCC. We further padded this data to make it constant in length to serve as an input for the audio models
- Audio embeddings using Wav2vec2 - Here we use wav2vec2 model which is pretrained to generate features from the audio data.

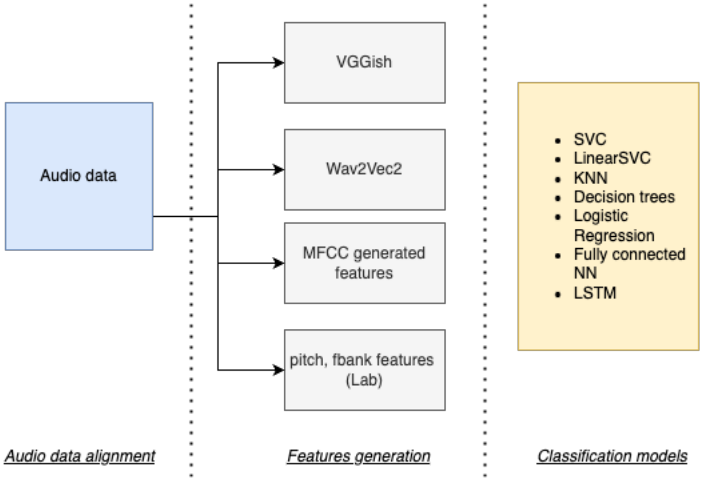

Figure 3: Audio for DAC

3.3.3 Audio model implementation. We use a combination of different embeddings/features with multiple simple models, deep learning models, and transformer-based models, similar to textual data for performing classification. They are :

- Simple models: SVM, Logistic Regression, K-neighbors classifier, Decision trees
- Deep learning model: Fully connected neural networks with hidden layers, LSTM models

[2] https://towhee.io/towhee/audio-embedding-vggish
[3] https://hub.towhee.io/towhee/torch-vggish

Table 2: Transcriptions vs dialogue act utterances example from conversation id - 2228: DA denotes Dialogue Act

Text in transcription data	Dialogue act utterances	DA label	DA
um-hum no i don't think it was really corruption that caused it	Uh-huh	aa	Agree/Accept
	No,	aa	Agree/Accept
	I don't think it was really corruption that caused it	sv	Statement-Opinion

- Transformers: pre-trained transformer model Hidden unit - BERT (HuBERT). Features produced by wave2vec2 were properly padded using DataCollator and then fed into HuBERT model that is specifically pretrained for speech classification.

3.4 Multimodal model implementation

With this project, we aimed to explore the performance of multimodal models for DA classification using both textual and audio modalities. This is helpful since the speech features can provide additional information about the text data.

In light of the above, we aim to merge/concatenate the features of textual and audio modalities and using late and early fusion techniques to build multimodal models. Similar to the models employing text modality, we plan to implement several multimodal models as a comparative study.

3.4.1 Model Implementation. For multimodal model implementation, we performed early/late fusion with simple and deep models, using audio embeddings from VGGish and other audio features such as fbank, pitch, etc.

The models are as follows:

- Simple models: SVM, Logistic Regression, XGBoost, Perceptron, using Late Fusion.
- Deep learning model: FNN model with early fusion. 2 different FNN networks were designed. One just performed concatenation of audio and text features. The other model focused on passing audio and text features through individual linear layers, before concatenating and passing through a final linear layer.
- Late Fusion: Probabilities for all dialogue acts were obtained from both audio and text models and then were added together. The resulting probabilities were then used to get the DA for the utterance.

4 RESULTS

We use accuracy and weighted F1 score as our performance metrics.

Table 3 shows the performance of simple models like SVM, Logistic Regression, etc with TF-IDF Vectorizer. From the table, we can see that XGBoost outperforms the other models in terms of accuracy and F1 score, especially on the data with only lowercasing and expanding contractions pre-processing with an accuracy of 71.30% and F1 score of 67.95%. Perceptron has the lowest performance among the models on both types of data. The additional pre-processing of non-alphabetic character elimination does not seem to improve the performance of the models.

Table 3: Simple models with TF-IDF Vectorizer: LR denotes Logistic Regression

	Lowercasing and Expanding Contractions		Lowercasing, Expanding Contractions, and non-alphabetic character elimination	
Model	Accuracy	F1 Score	Accuracy	F1 Score
SVM	69.45	65.86	68.7	65.23
LR	70.1	66.65	69.6	66.8
XGBoost	**71.3**	**67.95**	70.6	67.2
Perceptron	62.8	61.77	60	59.6

Table 4: BiLSTM Models with Glove Embeddings: 1L/2L denotes number of birdirectional LSTM layers. Each BiLSTM layer has 50 units.

	Glove Twitter 100d		Glove Wikipedia 100d	
Model	Accuracy	F1 Score	Accuracy	F1 Score
BiLSTM (1L)	**74.62**	**77.03**	73.43	76.51
BiLSTM (2L)	74.23	76.67	72.95	75.87

Table 5: Models with RoBERTa Features

Model	Accuracy	F1 Score
SVM	75.91	73.12
LR	75.48	72.57
XGBoost	74.57	71.15
MLP (1L)	77.18	78.10
MLP (2L)	**78.93**	**80.97**
MLP (3L)	75.03	76.23

Table 6: Context-Aware Model Results

Features	Accuracy	F1 Score
BERT	80.69	81.96
RoBERTa	**81.98**	**83.11**

With deep-learning models like Bidirectional-LSTM, we can see in Table 4 that we get much better performance. From this table, we can see that the BiLSTM model with one layer outperforms the model with two layers on both types of embeddings in terms of accuracy and F1 score of 74.62% and 77.03%, respectively. Additionally, the models using Glove Twitter 100d embeddings outperform those using Glove Wikipedia 100d embeddings.

Table 7: Models with BERT Features

	Lowercasing and Expanding Contractions		Lowercasing, Expanding Contractions, and non-alphabetic character elimination		Keeping case and expanding contractions	
Model	Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score
SVM	74.57	72	73.73	70.23	74	70.77
LR	74.04	71.15	71.64	68.94	72.53	69.72
XGBoost	72.62	69.73	70.87	69.2	70.62	68.73
MLP (1L)	75.84	76.67	74.04	75.13	74.45	75.69
MLP (2L)	**77.47**	**79.23**	75.49	76.17	76.24	77.47
MLP (3L)	73.29	75.67	73.18	75.21	73.87	75.62

Table 8: Audio modality models results

	fbank,pitch (max pooling)		fbank,pitch (90percentile pooling)		fbank,pitch (95percentile pooling)		VGGish	
Models	Accuracy	F1 score	Accuracy	F1 score	Accuracy	F1 score	Accuracy	F1 score
LR	45.9	30.6	43.8	28.6	44.3	28.9	47.9	32.5
Linear SVC	35.4	29.4	32.6	27.9	33.6	28.5	5.1	8.1
DT	28.4	28.1	29.1	29.1	28.1	28.1	26.4	27.1
KNN	47.1	32.7	46.1	33.6	45.7	32.4	**48.3**	31.5
FNN	47.5	32.1	45.8	31.3	46.14	31.67	**48.4**	31.5

Table 9: Deep FNN multimodal model with ROBERTa text features and audio features

	With audio features		With VGGish audio embeddings	
Model	Accuracy	F1 Score	Accuracy	F1 Score
FNN Early Fusion	42.10	42.09	30.72	30.72
FNN Early Fusion with individual linear layers	48.35	48.34	48.35	48.34

Table 10: Simple multimodal models with RoBERTa text features and audio features

	With audio features such as pitch		With VGGish audio embeddings	
Model	Accuracy	F1 Score	Accuracy	F1 Score
SVM	**72.54**	**69.44**	48.34	31.51
LR	71.54	69.62	43.24	33.33
XGBoost	69.19	66.99	46.24	31.02
Perceptron	46.08	51.67	48.64	34.80

Table 11: Multimodal model with late fusion

Model	Accuracy	F1 Score
RoBERTa + VGGish embeddings	**66.36**	**66.36**

However, with the current state-of-the-art transformer-based models, we experimented with BERT and RoBERTa CLS token features. We got the best results using RoBERTa CLS token features, shown in Table 7 and Table 5. Table 7 shows the performance of various models using BERT features for three different text preprocessing methods: lowercasing and expanding contractions, lowercasing and expanding contractions with non-alphabetic character elimination, and keeping the case and expanding contractions. The models include SVM, Logistic Regression, XGBoost, and Neural Networks with 1, 2, or 3 layers. Table 5 shows the results of using RoBERTa features. For RoBERTa, we only experiment with lowercasing and expanding contractions.

Using RoBERTa features, the Neural network with two layers on top of RoBERTa CLS token features performed the best across all pre-processing methods with an accuracy of 78.93% and F1 score of 80.97%. SVM, Logistic Regression, and XGBoost had slightly lower accuracy and F1 scores but still performed reasonably well.

Further, Table 6 shows that the context-aware model using RoBERTa as a feature encoder performed the best with an accuracy of 81.98% and an F1 score of 83.11%. This means that adding information about the context and speakers helps in a better understanding of the current utterance.

The results for unimodal models using audio modality can be seen in Table 8. This shows the performance of audio features generated by first and second methods stated in section 3.3.2. We use max, mean , 90 percentile and 95 percentile functionals on the data to perform pooling to provide input to the models. We observe that the results are quite comparable in all the cases. For the case with mean pooling we observed lower accuracies and hence did not report it. We observe that we both K-nearest neighbors classifier (KNN) and Forward neural network (FNN) perform the best compared to other models. Our FNN consists of 4 hidden layers of relu network with dropout rate of 0.2 and optimised using Adam optimizer. We found this to be provide the best accuracy nearing 47% for fbank and pitch features using max pooling method. We further also used VGGish network to generate features for audio and observe that these features perform better comparatively providing an accuracy of 48% using logisitic regression, KNN and Fully connected neural network.We created a similarly modelled FNN as above experimenting with different number of neurons in each hidden layer.

We further implemented LSTM and HuBERT model with features produced using MFCC and wav2vec2 respectively as stated in last two methods in section 3.3.2. We observed that features produced by MFCC provided the best accuracy of 50.64% on using an RNN network made up of LSTMm which is very less compared to the accuracy provided by lexical models.

Table 10 shows the performance of simple multimodal model (with early fusion) such as SVM, Logistic Regression, XGBoost and Perceptron. These models were provided audio features such as pitch, fbank and audio embeddings from VGGish network. From the table, it is evident that SVM and Logistic Regression performs the best with simple audio features, nearly matching the performance of text based models. We believe textual features overpower the model, which could explain the performance. Perceptron model performs the worst with an F1 score of 51.67% on simple audio features and 34.80% on audio embeddings from the VGGish network.

We also employed deep neural models for multimodal model implementation, with early fusion, as demonstrated by Table 9. First model performs concatenation of the text and audio features which are then passed through a fully connected layer. Since we believed that text features were overpowering audio features, we planned on implementing a second deep multimodal model that first passes the text and audio features through a linear layer and the outputs of both are then provided as input to a third fully connected network. This gave us a performance boost in F1 scores from 30.72% to 48.34% for both the audio features. This performance seems to be limited by the performance of the audio models.

The final multimodal model implemented employed late fusion with RoBERTa text features and VGGish audio embeddings. Output probabilities from both the models were summed up to get output probabilities. This configuration provided the best performace out of all deep multimodal models, with a jump of 37% in performance metrics.

5 CONCLUSIONS AND LESSONS LEARNED

From the results, it is very evident that text based models, especially by providing the context have performed the best. In case of unimodal models based in audio features we see that features generated using MFCC on LSTM model provide the best accuracy but it is not comparable to the accuracy obtained using lexical models.

In case of multimodal models, the performance has not been upto the levels of text based models and is very much limited by the performance of audio models. Comparing early fusion and late fusion implementations, late fusion seems to yield the best performance out of all multimodal models.

All in all, text features from RoBERTa models have proven to be quite essential for DA classification and the performance numbers seems to dictate that multimodal model perform quite poorly and is limited by the performance of audio models. However, audio files and related features provides more information about the utterance such as intonation, pitch which can complement the text models well. The aim going forward is to keep improving the audio models, through which we expect better performance in multimodal settings.

6 CONTRIBUTIONS

- Geet Shingi: Data Preparation and Summary Statistics, Textual data classification, Context-Aware Model implementation, Literature Survey, Report
- Samhitha Gundam: Literature survey, Report ,Audio feature extraction using MFCC,Wav2Vec2, fbank and pitch features,Audio classification models implementation, Audio data preparation, Report
- Swapnil Arya: Literature Survey, Audio Transcript data cleaning, Audio Data Preparation, Audio Time Alignment, VGGish audio embeddings, Multimodal model implementation, Report

REFERENCES

[1] Ali Ahmadvand, Jason Choi, and Eugene Agichtein. 2019. Contextual Dialogue Act Classification for Open-Domain Conversational Agents. 1273–1276. https://doi.org/10.1145/3331184.3331375

[2] Lin Chen and Barbara Di Eugenio. 2013. Multimodality and Dialogue Act Classification in the RoboHelper Project. In *Proceedings of the SIGDIAL 2013 Conference*. Association for Computational Linguistics, Metz, France, 183–192. https://aclanthology.org/W13-4031

[3] Zheqian Chen, Rongqin Yang, Zhou Zhao, Deng Cai, and Xiaofei He. 2018. Dialogue Act Recognition via CRF-Attentive Structured Network. In *The 41st International ACM SIGIR Conference on Research amp; Development in Information Retrieval* (Ann Arbor, MI, USA) *(SIGIR '18)*. Association for Computing Machinery, New York, NY, USA, 225–234. https://doi.org/10.1145/3209978.3209997

[4] Zihao He, Leili Tavabi, Kristina Lerman, and Mohammad Soleymani. 2021. Speaker Turn Modeling for Dialogue Act Classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 2150–2157. https://doi.org/10.18653/v1/2021.findings-emnlp.185

[5] Hamed Khanpour, Nishitha Guntakandla, and Rodney Nielsen. 2016. Dialogue Act Classification in Domain-Independent Conversations Using a Deep Recurrent Neural Network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, Osaka, Japan, 2012–2021. https://aclanthology.org/C16-1189

[6] Harshit Kumar, Arvind Agarwal, Riddhiman Dasgupta, and Sachindra Joshi. 2018. Dialogue Act Sequence Labeling Using Hierarchical Encoder with CRF. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence* (New Orleans, Louisiana, USA) *(AAAI'18/IAAI'18/EAAI'18)*. AAAI Press, Article 421, 8 pages.

[7] Ji Young Lee and Franck Dernoncourt. 2016. Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, 515–520. https://doi.org/10.18653/v1/N16-1062

[8] Ruizhe Li, Chenghua Lin, Matthew Collinson, Xiao Li, and Guanyi Chen. 2019. A Dual-Attention Hierarchical Recurrent Neural Network for Dialogue Act Classification. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Association for Computational Linguistics, Hong Kong, China, 383–392. https://doi.org/10.18653/v1/K19-1036

[9] Yang Liu, Kun Han, Zhao Tan, and Yun Lei. 2017. Using Context Information for Dialog Act Classification in DNN Framework. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 2170–2178. https://doi.org/10.18653/v1/D17-1231

[10] Daniel Ortega and Ngoc Thang Vu. 2017. Neural-based Context Representation Learning for Dialog Act Classification. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*. Association for Computational Linguistics, Saarbrücken, Germany, 247–252. https://doi.org/10.18653/v1/W17-5530

[11] Tulika Saha, Aditya Patra, Sriparna Saha, and Pushpak Bhattacharyya. 2020. Towards Emotion-aided Multi-modal Dialogue Act Classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 4361–4372. https://doi.org/10.18653/v1/2020.acl-main.402

[12] John R. Searle. 1969. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, London.

[13] Sheng-syun Shen and Hung-yi Lee. 2016. Neural Attention Models for Sequence Classification: Analysis and Application to Key Term Extraction and Dialogue Act Detection. (03 2016).

[14] Trang Tran, Jiahong Yuan, Yang Liu, and Mari Ostendorf. 2019. On the Role of Style in Parsing Speech with Neural Models. In *Proc. Interspeech 2019*. 4190–4194. https://doi.org/10.21437/Interspeech.2019-3122