Augmented Speech Generation through Audio Signals

Arushi Jain
School of Information
University of Michigan
arushij@umich.edu

Sagnik Sinha Roy
School of Information
University of Michigan
sagniksr@umich.edu

Yixian Zhou
School of Information
University of Michigan
zyixian@umich.edu

Executive Summary

Today, dialogue agents in digital assistants (Siri, Alexa, Google Now/Home, Cortana, etc.), give directions, control appliances, find restaurants, or make calls and research on building such dialogue systems that can converse with humans naturally has recently attracted a lot of attention. So, it is very crucial to understand how humans converse with each other. We have seen that most work done in this area is based on text features i.e. only text data is used to make the response generation models. In contrast, human-human conversations involve other modalities, such as voice, facial expression and body language, which influence the conversation.

Deriving the inspiration from this thought, we tried to explore the impact of incorporating the audio features of the user into massively pre-trained language generation model - GPT-2 released by OpenAI [11]. We first tried to look for multi-modal datasets and finalised Multimodal EmotionLines Dataset (MELD) provided by Poria et. al.[10] as it contains audio and video files along with the text from the famous TV series - Friends. Next, we worked on audio feature extraction and selection. There are many audio features which can be representational for voice, tone, mood of the user. Eventually we ended up creating our own "audio embeddings" just like word embeddings which involved a lot of pre-processing of the data from extraction of wav files to generating a sequence of numbers and then trained a word2vec model on these sequences.

Next, we had 2 major tasks -classification and generation. For emotion and sentiment classifiers we employed different classification techniques and compared them with baseline models which were developed with only text features (audio-free models). We suffered with high imbalance problem due to 'Neutral' class in both the classifiers which was treated by setting class weights. As a result, our sentiment classification model showed 55% accuracy and F-1 score of 54% as compared to the baseline model with 52% accuracy and 50% F-1 score. Similarly, emotion classification model showed 45% accuracy and F-1 score of 45% as compared to the baseline model with 44% accuracy and 43% F-1 score.

For generation, we first incorporated the audio features in the transformer network but it was performing worse than the baseline. So, we decided to augment our model with audio features by helping in choosing the best generated response only. We compared audio embedding of the input test sample and the audio embeddings for each of the 5 sample responses generated and kept the one with the highest cosine similarity. Our final model is able to achieve a lower perplexity (198.89) as well as a higher BLEU score (0.4489) as compared to the baseline model (perplexity - 284.26 and BLEU score - 0.1847).

Lastly, we conducted survey on Amazon Mechanical Turk for human evaluation and found that 20% of the time our model was successful to confuse the humans whether the reply is machine generated or actually uttered by a human being which showed that we were able to enhance the model by utilising audio features but it is not of human-like quality.

1 Introduction and Motivation

With the advancement in the field of NLP, chatbots have become really popular and useful in recent times. Most of them implement dialogue systems that work on text data extracted from audio conversations. This setting, however, is oversimplified compared to real-world human conversation, which is naturally a multimodal process. Information can be communicated through voice, body language and facial expression. In some cases, the same words can carry very different meanings depending on information expressed through other modalities [14].

Moreover, with the increasing use of conversational bots like Alexa and Google Home day by day, we notice that they do not sense emotions as people also convey emotions and sentiments in conversations by changing their accent, tone, and speed etc. They maybe identical from a written point of view, utterances may acquire different meanings based solely on audio information. Empowering a dialogue system with such information is necessary to interpret an utterance correctly and generate an appropriate response. In other words, audio contains useful information such as emotions and sentiments which text cannot capture and dialogue systems may generate template-based and repetitive responses without noticing the change in people's voice.

In this paper, we present a speech generation system using both text and audio signals. First, we tried different techniques to extract audio features detailed in feature selection section. Then, we built classifiers which detect sentiment and emotion through text data and audio files. Finally, we built a language generation model which incorporates sentiments, emotions along with textual features under Transformer framework. We augmented this generation by taking into account audio signals, we select the best generated sentence based on the cosine similarity of audio embeddings of the input sentence and the generated sentence.

2 Related Work

Majumder et al.[7] introduced DialogueRNN which is the current state of the art of detecting emotions in conversation. DialogueRNN uses gated recurrent units to track speakers' states for classifications. Attending over this GRU gives contextual representation that has information of all preceding utterances by different parties in the conversation. Wollmer et. al. [13] used contextual information for emotion recognition in multimodal setting. Recently, Poria et al. [9] successfully used RNN-based deep networks for multimodal emotion recognition, which was followed by other works with the same approach.

Similar to this, there a number of approaches to multimodal sentiment analysis as well. Actually, sentiment analysis can be performed at different granularity levels, e.g., subjectivity detection simply classifies data as either subjective (opinionated) or objective (neutral), while polarity detection focuses on determining whether subjective data indicate positive or negative sentiment. Emotion recognition further breaks down the inferred polarity into a set of emotions conveyed by the subjective data, e.g., positive sentiment can be caused by joy or anticipation, while negative sentiment can be caused by fear or disgust. Poria et al. [1][2][6] extracted audio, visual and textual features using convolutional neural network (CNN); concatenated those features and employed multiple kernel learning (MKL) for final sentiment classification. These works formed the basis of our approach that sentiment analysis and emotion detection are also very important while generating responses and there have been many multimodal studies regarding same. Human conversation itself involves multiple channels of information. Voice, body language and facial expressions all play an important role in conversation. In an ideal human-machine conversational system, machines should be capable of understanding these multimodal language cues. In the literature, this information has seen use in conversation analysis. Yu [15] proposed to model user engagement and attention in real time by leveraging multimodal human behaviors, such as smiles and speech volume. Gu et al. [4] performed emotion recognition, sentiment analysis, and speaker trait analysis on conversation data using a hierarchical encoder that formulates word-level features from video, audio, and text data into conversation-level features with modality attention.

Very recently in Jan 2020, in contrast to the popular text-only assumption Young et al.[14] extracted features from audio and incorporated them in their dialogue generating system which outperformed other audio-free models in perplexity and human evaluation. This is the only paper which by far which

has used audio features of the user message in neural conversation generation and it outperforms the baseline audio-free model in terms of perplexity, diversity and human evaluation.

These result motivates our work – since incorporating audio features improves emotion classification, sentiment classification in conversation and they are important to response generation, we hypothesize that incorporating emotion and sentiment features alongwith augmenting it audio features for selecting the best response improves response generation.

3 Dataset

We used the recently released Multimodal EmotionLines Dataset (MELD) provided by Poria et. al.[10] MELD contains audio and video files along with the text from the Friends TV series, which has 13,708 utterances from the show. Each utterance is labelled with emotion and sentiment as well. There are 7 labels including Joy, Sadness, Fear, Anger, Surprise, Disgust and Neutral and 3 sentiments including Positive, Negative and Neutral.

We performed exploratory data analyses on the MELD dataset, and some of our findings are shown here. We plotted the distribution of sentiment and emotion labels (Figures 1 and 2), and observed that for the sentiment labels, we actually have a huge class imbalance for the neutral sentiment. Similarly, for the emotion labels, we have an imbalance for the neutral emotion. This is why in our baselines, we used methods to handle this using class weights.

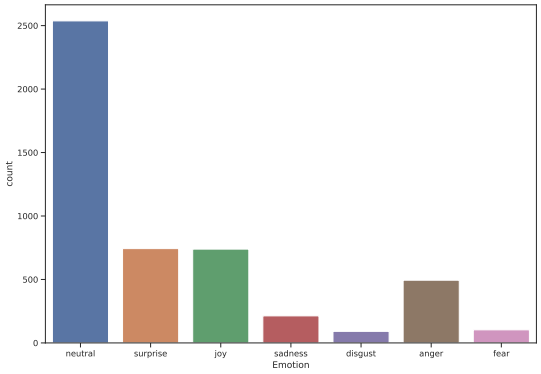

Figure 1: Bar plot showing count by emotion categories

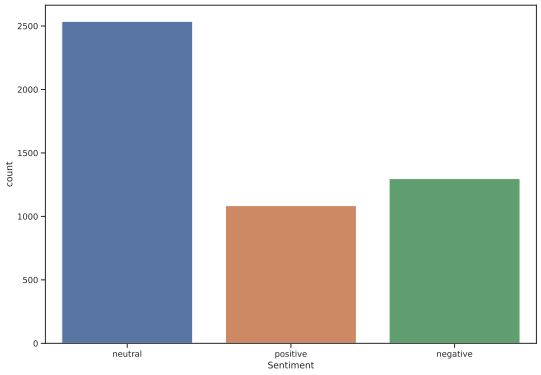

Figure 2: Bar plot showing count by sentiment categories

An interesting feature to understand the dataset is to visualize the duration of each conversation present in the dataset to formulate some basic ideas about how duration could correlate to sentiment or emotion, as duration could be one of the audio features. The data provided us with start and end times, which we used to calculate the duration, and we made various plots to understand the distribution of duration in the dataset (Figures 1, 2, and 3). We observe that most conversations are within 1-5 seconds. Further, an interesting observation we had was that while duration violin plots (Figure 4) for different sentiments are similar, the positive sentiment seems to have a longer tail, which means positive sentiment could have the longest conversations. Similarly, the joy emotion had the longest tail for the violin plots for emotions. We also made a plot for duration against emotions grouped by sentiment (Figure 5). This plot gives us the confidence that the sentiments are labelled accurately, as the positive and negative emotions are grouped accurately. Additionally, it gives us the intuition that the average duration for a negative sentiment seems to be higher.

For the audio and video data, we also experimented with ways to represent the audio as features, which is described in depth in section 4.1

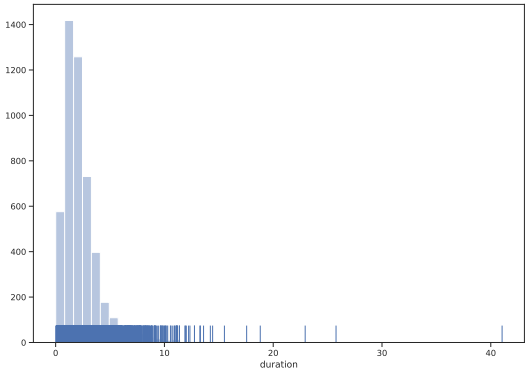

Figure 3: Histogram of the duration of conversations

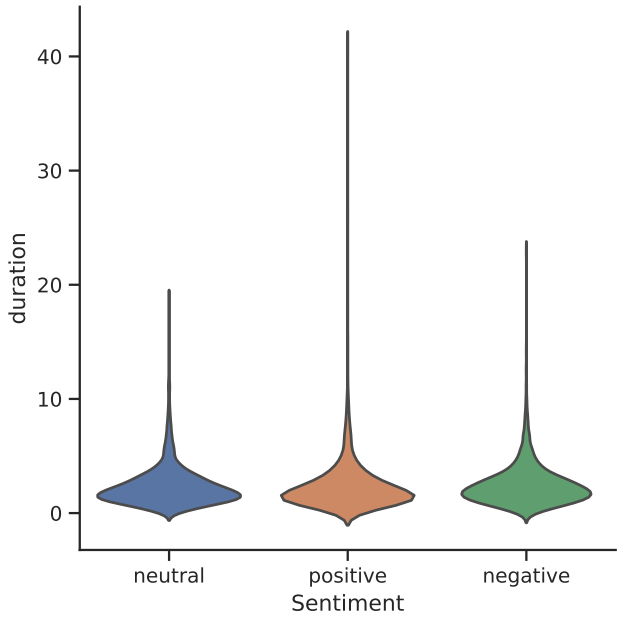

Figure 4: Violin Plots showing count by different emotion categories

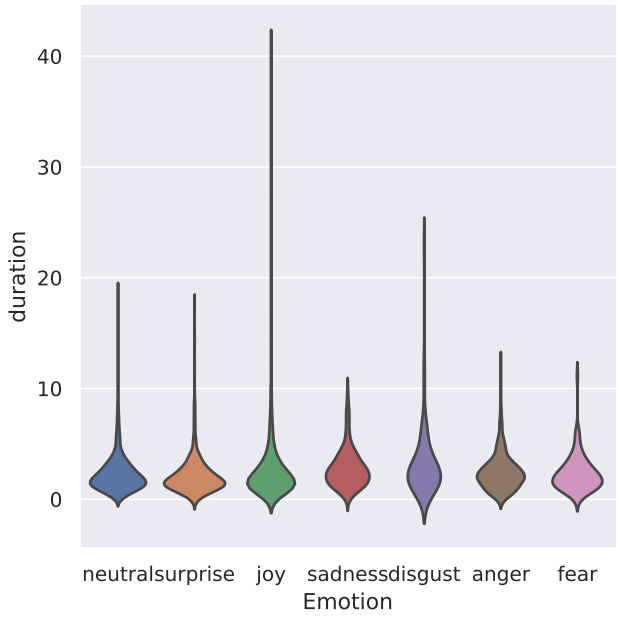

Figure 5: Duration with sentiment and emotion together

4 Methodology

4.1 Audio Features

We obtained certain audio features namely - Mel Spectrogram Frequency, Mel Frequency Cepstral Coefficient (MFCC) and chromagram for each sentence using a Python package librosa [8]. Spectrum is a common tool used in signal processing. It's a representation of a sound signal in frequency domain with Hertz scale. However, humans are better at identifying small changes in speech at lower frequencies, meaning that extract features based on Hertz scale may miss useful information in voice signals. Thus, we used Mel scale feature instead which is better at implementing the working principles of human's ear and really helpful for speech analysis. Mel scale spectrum is computed by taking the log of the magnitude of the Fourier transformation of a sound signal (in time domain). Another commonly used feature in speech recognition and classification is MFCC (mel frequency cepstral coefficient). It is computed by taking the spectrum of the Mel scale spectrum by a cosine transformation. Lastly, Chroma feature represents the tonal content (12 different pitch classes) of a musical audio signal.

4.1.1 Audio Embeddings

Previously we described popular audio features which we extracted using existing packages and in this section we will talk about our own extraction of audio features process. We applied the concept of word embeddings to the audio data to get "audio embeddings", i.e. n-dimensional vectors which represent each audio file. To perform this task, we first used the software FFMPEG to extract wav files (containing only audio) from of the given mp4 files which were video clips. Once we got the wav files, we used the python package wav2vec which parses a wav file and generates a sequence of frequencies across multiple channels to represent the audio information as sequences of numbers. Since it is a sequence representation of the audio, we argued that it can be treated similar to a sequence of words, where the magnitude of the pitch is similar to token representation of words. So, we extracted these audio sequences from all the audio files, and then trained a word2vec model on these sequences. This model was our approach to generate audio embeddings of 20 dimensions from any given wav file. We could not perform quantitative evaluation of this model due to time constraints but we did some qualitative analysis by randomly selecting wav files, and looking at it's nearest neighbors. For 3 out of the 5 samples, we observed that the nearest neighbor of a given audio file was part of the same

scene of the show as the original wav file. We used these embeddings to train our generation model, which is detailed in section 4.3

4.2 Classification

We employed different classification techniques to build sentiment and emotion classifiers, including Naive Bayes, Random Forest, Logistic Regression, LSTM, Multi-Layer Perceptron etc. To see how audio signals can improve our classifiers, we built baseline classifiers using only text features and enhanced classifiers using text and audio features. We used a bag of words approach with TF-IDF vectorization to get the text features and concatenated audio features with text features.

Given the fact that the data is distributed unevenly across classes, we experimented with several methods to tackle this challenge. We have tried Random Oversampling and SMOTE [3] for over-sampling and Random Undersampling and Edited Nearest Neighbors [12] for down-sampling. We also configured some algorithms to be cost-sensitive [5] to different classes, such as Logistic Regression and Random Forest. By setting the class weights, we were able to modify the cost of misclassification to be inversely proportional to the distribution of the training dataset. Among all these methods, we found that the cost-sensitive models had the best performance.

4.3 Speech Generation

To perform the task of speech generation, first, we modified the data structure to change them into "Input" and "Reply" format, that is for each of the conversations, the first utterance by a character would be the input, and the next utterance by a different character would be the reply. This reply would then be the input for the next data point, and the utterance after that would be the reply and so on. This is the data format we used to generate replies from any given input statement.

For the baseline, we used a massively pretrained model called GPT-2 which was recently released by OpenAI [11]. This model is based on the "Transformer" based neural network, which was found to perform really well for text generation. Initially, we used this pretrained model directly to generate replies, and calculated the perplexity and BLEU scores of the generated utterances. However, we saw absurdly high values of perplexity, and really low values (of the order of 10-34) for the BLEU scores. This was too poor, even for a baseline.

Therefore, we fine-tuned the pre-trained model for the MELD corpus. This was achieved by creating a corpus of sentences out of MELD, which acts like the filtered language model, and then we run the training process on the existing weights for this corpus to create a specialized language model which is purely for the MELD corpus. We then performed the task of generation again, and saw much better results for both, the perplexity values and the BLEU scores. The results for these are outlined in the Results section.

For the actual generation task, first, we implemented an LSTM network. To represent the text features, we also trained a Word2Vec model on the text of the corpus. For the embedding of the input utterance for each layer, we used a linear combination of the text vectors and audio vectors as shown in Figure 6. An interesting note here is that the text features of the model for each utterance are at a word level, as we have a vector representation for each word. However, the audio features are at utterance level, as we have one vector for each wav file. To successfully merge these two features, we used the same audio embeddings for each word of any given utterance. Our reasoning was that since any given utterance has similar background noise, the contribution at utterance level would be very similar to that in word level. Once our data structure was ready, we trained the LSTM network on the dataset, using the adam optimizer function, and the sparse categorical cross-entropy loss function. After 300 epochs, we noticed a loss of barely 0.9018. However, when we calculated the average perplexity and BLEU scores of this language model, we observed that it was performing worse than the baseline. Therefore, we decided to stick to the Transformer architecture.

Finally, we implemented a Transformer network. However, we did not find a convenient way to integrate the audio embeddings or features into this network. Therefore, instead, we decided to generate multiple samples from a given utterance using only the text features (and some categorical audio features), and then use the audio embeddings to select the best generated sample. To ensure high quality generation, we formatted the dataset with special tokens indicating start and end of utterance, as is consistent with training dialogue generation models. Along with those, we also

included special tokens for emotion and sentiment labels into the data. For example, the anger emotion was represented with the <anger> token, and positive sentiment was represented with the <sp> token. This modified dataset was then used to train the network.

Once the training was complete, to test the generation quality, we generated 5 samples for each test utterance. For each generated sample, we queried for the closest match of tokens from the training dataset. Let's call these 5 matched utterances as matches. We checked the cosine similarity between the audio embedding of the input test sample and the audio embeddings for each of the matches. The match with the highest cosine similarity resulted in the selection of the corresponding generated sample as the final generated result for that particular test datapoint.

Figure 6: Input Embeddings of LSTM Network

We calculated the language metrics for this model, and were pleased to see that we were able to beat the baseline. Further, we have also created a survey for speech generation results which compares several examples of human and bot utterances. Some of the generated samples of our project can be seen in table 3

All the materials utilized and code developed in this project can be found on this Github Repository: https://github.com/scarescrow/AuSpeech

5 Results

5.1 Classification

Since we are predicting class labels, we used classification accuracy as one of the metrics for the classification tasks. As we mentioned in the Dataset section 3, there is a huge imbalance distribution across classes of both sentiments and emotions, more than half of the utterances are labelled as Neutral (majority class). Considering that we want to identify as many classes (e.g. Negative, Anger, Joy etc.) as possible to give personalized responses, we chose F-Measure to optimize the precision and recall. We used the F1 score as another metric as we considered false negative and false positive equally important.

One can see from the Table 1 that accuracy goes down when we use class weights as compared to those models where we do not use class weights. But we purposely included class weights due to high imbalance of the classes. We saw that Logistic regression showed the best accuracy for both emotion as well as sentiment classification. We also used cross-validation to tune the hyper-parameters.

After experimenting with different audio features, we finalized our classification models which has two kinds of audio features, Mel Frequency Cepstral Coefficient and Chroma along with text features. According to Table 1, although we achieved higher accuracy score in emotion classification using audio embeddings, the classifier mainly predicted the label of majority class (Neutral) and failed in identifying other minority classes, which resulted in lower F-1 score. So, we decided to include only audio features and not audio embeddings in both the classifiers to be consistent.

5.2 Speech Generation

To evaluate the quality of generation for both the baseline model and the audio enhanced model, we used two most commonly used metrics to evaluate language models: BLEU score and perplexity. We choose these scores, as they can give a decent idea about how good the language model is, specific to the corpus.

The comparison of the metrics for the baseline speech generation model with the final model are outlined in Table 2. Our final model is able to achieve a lower perplexity as well as a higher BLEU score as compared to the baseline model. However, it should be noted that even the final metrics

		Technique	Accuracy	F-1 score
Sentiment Classification	Baseline (text)	**Logistic Regression (with class weights)**	**52%**	**50%**
		Logistic Regression (without class weights)	54%	48%
		Vader Sentiment Analyzer (NLTK)	48%	60%
		Random Forest (without class weights)	50%	45%
		Random Forest (with class weights)	48%	48%
	Our Model (audio and text)	**Logistic Regression (audio features using MFCC and Chroma)**	**55%**	**54%**
		Logistic Regression (audio features using embeddings)	51%	50%
Emotion Classification	Baseline (text)	**Logistic Regression (with class weights)**	**44%**	**43%**
		Logistic Regression (without class weights)	51%	45%
		Multinomial Naive Bayes	51%	42%
		Random Forest (without class weights)	47%	30%
		Random Forest (with class weights)	23%	24%
	Our Model (audio and text)	**Logistic Regression (audio features using MFCC and Chroma)**	**45%**	**45%**
		Logistic Regression (audio features using embeddings)	47%	37%

Table 1: Comparison of Classification between Baseline model and Audio-Enhanced model

Model	Perplexity	BLEU
Baseline	284.26	0.1847
Audio-Enhanced	198.89	0.4489

Table 2: Comparison of Metrics between Baseline model and Audio-Enhanced model

are nowhere close to an actual generation model in today's world. Part of the reason for this is the limitation in scope of the dataset. The metrics do not produce an accurate outlook of the actual generation quality. To mitigate this limitation, we conducted a human evaluation, which is outlined in the following section. Table 3 below shows some examples generated by our model:

Context	Actual Reply	Generated Reply
hey, did you pick a roommate?	you betcha!	yeah i did. We just made plans for monday!
nobody respects the bucket	you wouldn't believe what people put in here	but remember, this is not mine, it is for yours
i mean what's more important?	what people think or how you feel, huh?	always thinking about building it and then moving forward to building the next!

Table 3: Generated Samples from Audio-Enhanced model

5.3 Human Evaluation

Apart from the results shown above, we also developed a survey using Amazon Mechanical turk for human evaluation. We randomly selected 200 different responses generated from our model and compared them against actual responses. We choose 3 as our number of respondents for each HIT so in total we had 600 responses where turkers were given a prompt as shown in Figure 7. We gave them a sentence uttered just before the response generated as Person 1 and then 2 respnses- one actual and one generated from our model as Person 2 asking them to choose one which sounds more human-like.

We found that 80% of the time they identified the actual reply as human-like but rest 20% of the time they choose the responses generated by our model as more human-like. This means that 20% of the time our model was successful to confuse the humans whether the reply is machine generated or actually uttered by a human being. Also, we calculated krippendorff's alpha to see the disagreement between raters. We found that krippendorff's alpha = 0.4864 which means there is moderate agreement between annotators.

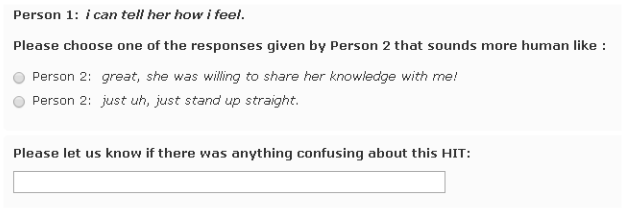

Figure 7: Example from the Survey conducted on Amazon Mechanical Turk

6 Conclusion

In this work, we augmented the transformer dialogue generation model with audio features and showed that the resulting model outperforms the audio-free baseline on several evaluation metrics. However, we are still far behind when it comes to human evaluation because 80% of the time humans could identify the actual and machine generated results. So, although we could enhance the model by utilising audio features but it is not of human-like quality.

For extracting audio features, we learnt the biggest lesson that one should not start with cumbersome and hard to implement techniques, one should always start simpler. We first took harder approach to extract audio features using OpenSmile software but later realised there are easier ways like using libraries and developing our own audio embeddings. There are handful of papers which have utilised audio features in generating responses, so it became more challenging for us to integrate audio features to transformer network.

Finally, we succeeded in our initial aim to integrate the audio features into transformer network with text, emotion and sentiment features but results generated were poorer than baselines which led us to modify our approach a bit. In the current model, we use audio features only to select the best generated sentence by comparing the cosine similarity of the audio embeddings which outperforms the text-only model. This proves that using text in dialogue systems is a good-enough approximation in a lot of scenarios but other modalities (i.e., video and audio) have to be integrated before automatic dialogue systems can reach human performance. Our research belongs to this paradigm of building multimodal chatbot systems through AI.

References

[1] Erik Cambria, Devamanyu Hazarika, Soujanya Poria, Amir Hussain, and RBV Subramanyam. Benchmarking multimodal sentiment analysis. In *International Conference on Computational Linguistics and Intelligent Text Processing*, pages 166–179. Springer, 2017.

[2] Erik Cambria, Soujanya Poria, Rajiv Bajpai, and Björn Schuller. Senticnet 4: A semantic resource for sentiment analysis based on conceptual primitives. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers*, pages 2666–2677, 2016.

[3] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[4] Yue Gu, Xinyu Li, Kaixiang Huang, Shiyu Fu, Kangning Yang, Shuhong Chen, Moliang Zhou, and Ivan Marsic. Human conversation analysis using attentive multimodal networks with hierarchical encoder-decoder. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 537–545, 2018.

[5] Gary King and Langche Zeng. Logistic regression in rare events data. *Political analysis*, 9(2):137–163, 2001.

[6] Navonil Majumder, Soujanya Poria, Alexander Gelbukh, and Erik Cambria. Deep learning-based document modeling for personality detection from text. *IEEE Intelligent Systems*, 32(2):74–79, 2017.

[7] Navonil Majumder, Soujanya Poria, Devamanyu Hazarika, Rada Mihalcea, Alexander Gelbukh, and Erik Cambria. Dialoguernn: An attentive rnn for emotion detection in conversations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6818–6825, 2019.

[8] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, 2015.

[9] Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Majumder, Amir Zadeh, and Louis-Philippe Morency. Context-dependent sentiment analysis in user-generated videos. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 873–883, 2017.

[10] Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. Meld: A multimodal multi-party dataset for emotion recognition in conversations. *arXiv preprint arXiv:1810.02508*, 2018.

[11] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[12] Dennis L Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, (3):408–421, 1972.

[13] Martin Wöllmer, Angeliki Metallinou, Florian Eyben, Björn Schuller, and Shrikanth Narayanan. Context-sensitive multimodal emotion recognition from speech and facial expression using bidirectional lstm modeling. In *Proc. INTERSPEECH 2010, Makuhari, Japan*, pages 2362–2365, 2010.

[14] Tom Young, Vlad Pandelea, Soujanya Poria, and Erik Cambria. Dialogue systems with audio context. *Neurocomputing*, 2020.

[15] Zhou Yu. Attention and engagement aware multimodal conversational systems. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 593–597, 2015.