Multimodal Sarcasm Detection in Sitcoms

Owen Lloyd, Garrett Parrish, and Anastasia Vopelius

***Abstract*—Sarcasm detection is a subset of sentiment analysis that looks to determine if the sentiment of some piece of content is sarcastic. There are a wide variety of sarcasm detection and sentiment analysis models, ranging from models trained on text alone to multimodal systems. Multimodal sentiment analysis looks to consider multiple different modalities (i.e. text, speech, visual) from the given piece of content to help determine the sentiment. We look to explore how the application of multimodal sentiment analysis to common American sitcoms could improve upon standard text-based sarcasm detection systems. Here we show that a bimodal model trained on text-based and visual features can outperform standard text-based sarcasm detection systems. This result confirms previous work that has found improved results in sentiment analysis through the inclusion of multiple modalities. It also reveals interesting relationships between each of the modalities, providing insight into which aspects provide the most predictive value when combined with others. We see that an increase in pitch is a strong indicator for sarcasm in audio data, and that performance increases when more context is given for the current piece of content being analyzed. We believe this paper to be a strong initial exploration of multimodal sentiment analysis applied to sarcasm detection, and see room for further work in the exploration of context in different settings as well as in the feature extraction portion of pipeline.**

I. Introduction

Sentiment analysis is a field focused on determining the underlying sentiment of various forms of unstructured content, typically in the form of text. Positive and negative sentiment analysis is an extremely well-studied field, with models able to predict, with near 98% accuracy, the sentiment of a given piece of text. Sarcasm detection is a very narrow part of the overarching field of sentiment analysis, focused on determining if unstructured content is being said in a sarcastic or more serious tone. Detecting sarcasm with purely text-based data is extremely difficult due to the nuances and figurative phrases that tend to indicate sarcasm. It is possible to aid these text-based methods by leveraging the context in which they are said to give an indication of their sarcastic content. Therefore the overall purpose of this paper is to improve sarcasm detection methods through the implementation of multimodal sentiment analysis techniques. As opposed to attempting to detect sarcasm solely based on text-based data, we will train our model on audiovisual clips. This will allow our model to learn to make classifications based not only on the contents of the text but the tone of the speaker's voice and as well as their face and body language. Our expectation is that we will be able to drastically improve the performance of our sarcasm detection by including the audio and visual clues compared to if we exclusively trained a model with the text data.

As previously mentioned, sentiment analysis as a whole is an extremely well-studied field with extremely effective models for determining positive and negative sentiments. Sarcasm detection, on the other hand, is a small subset of sentiment analysis that does not perform at the same level. Advancements in sarcasm detection would be a large step forward in the field of sentiment analysis. Additionally, it is extremely important to be able to understand and determine both the figurative and literal meanings of an individual's opinions, especially on social media. Beyond sentiment analysis specifically, improved methods of sarcasm detection will allow companies to have an easier time analyzing and understanding their customer's feelings about their products to improve their product quality.

During this paper, our team is working with audio, visual, and text-based data. All three of these are forms of unstructured data, making them much more difficult to analyze than structured data. There are multiple preprocessing steps necessary to transition from our raw audio, visual, and text-based data to a form of data that the models we choose to implement can interact with. Another challenge related to this work is the limited number of datasets available for multimodal sentiment analysis. The main datasets at our disposal, for multimodal sentiment analysis, are CMU-MOSEI (Multimodal Opinion and Emotion Intensity), MELD (Multimodal EmotionLines Dataset), and CMU-MOSI (Multimodal Corpus of Sentiment Intensity). This lack of data could potentially present challenges when it comes to the validation of our model and exploring the performance of our model on a dataset it was not trained on. Another challenge we have faced is determining the best model to train our audio, visual, and text-based data once we had it preprocessed and ready to use. While there are a few fusion methods such as Cross-Modal BERT that attempt to perform text-audio sentiment analysis, we have run into issues finding or developing a model that performs text-audio-visual sentiment analysis. Because of this, it is likely we will need to concatenate between multiple different models, each one designed for and trained on a specific part of our dataset (the audio, visual, and textual components).

II. Related Works

Sentiment Analysis

As previously stated, sentiment analysis is a field of study that typically utilizes natural language processing (NLP) and techniques related to text analysis to extract subjective information from text [2]. An increased focus on research related to sentiment analysis has evolved due to an increase in public available opinionated writing such as blogs. The field of

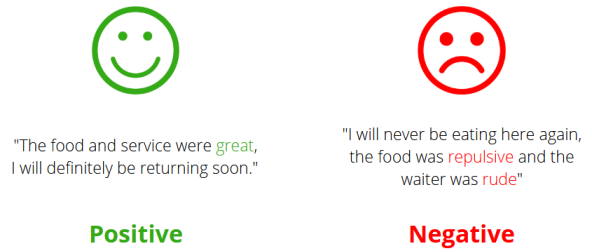

Fig. 1: Basic example of text-based sentiment analysis

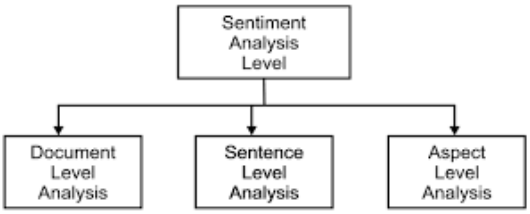

Fig. 2: Breakdown of the levels of sentiment analysis in early research

sentiment analysis has recently begun to also use other types of data such as audio and video. Sentiment analysis is important in aiding a business' ability to process and analyze product sentiment, commonly acquired through customer feedback, in a timely automated fashion. Sentiment analysis is a well-studied field with an abundance of research papers and models published showing the advancements in the field. Some recent applications of sentiment analysis have been directed towards political agendas, marketing campaigns, and social media platforms [2].

Research related to sentiment analysis first began around the turn of the century, primarily focused on analyzing subjectivity at the sentence level [2]. Early study of sentiment analysis can be broken down into three categories: document-level, sentence-level, and aspect-level sentiment analysis as shown in Fig. 2. below [2].

In document-level sentiment analysis, the goal is to extract information to determine the overall sentiment of the document. Sentence-level sentiment analysis instead focuses on the sentiment of individual sentences. Finally, aspect-level sentiment analysis refines the analysis even further to an entity-to-entity basis. While these three levels of granularity were the primary focus of early research related to sentiment analysis, there had also been research focused on the analysis of phrases within text, called phrase-level sentiment analysis [2]. Early research related to phrase-level sentiment analysis contributed to a rule-based sentiment analysis approach. Rule-based sentiment analysis computes sentiment based on a set of defined rules. A key component of rule-based approaches is sentiment lexicons, collections of words, and their associated polarity in terms of sentiment alignment [2]. Lexicons are resourceful in storing the polarity of certain words or phrases, but can often provide challenges when it comes to subjectivity when combined with rule-based approaches. This has lead to

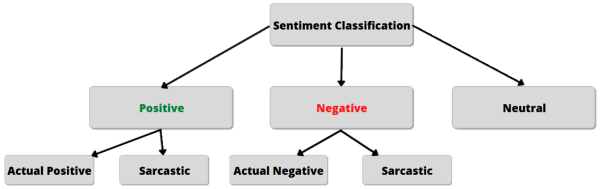

Fig. 3: Simple application of sarcasm detection on an existing sentiment classification framework

advancements in machine-learning based sentiment analysis approaches, such as combining Support Vector Machines or Naive Bayes Classifiers with lexicons, which tend to handle generalization better by not relying on manually defined rules [2].

More recently, research has focused on the use of deep learning to perform sentiment analysis in relation to natural language processing (specifically feature extraction). Convolutional neural networks and recurrent neural networks are two deep learning architectures that have been applied to the task of sentiment analysis [2]. Due to a reliance on annotated data, lexicons have been combined with deep learning approaches, to improve on many machine learning approaches. Many deep learning models such as BERT and XLNet have been proved to achieve state-of-the-art performance on numerous benchmark datasets used to analyze the effectiveness of the sentiment analysis model.

Sarcasm Detection on Text-Based Data with Contextual Clues

Although there has been an abundance of research conducted on sentiment analysis and many models have achieved state-of-the-art performance with exceptional accuracy, there are still issues in generalizing these models to handle data obtained from social media platforms such as Twitter. Additional processing is necessary for data obtained from these social media platforms (commonly referred to as micro-blogs). Posts on these platforms don't always follow the same conventions as most sources of texts such as books. For example, slang words and abbreviations are extremely common in these posts [2]. There is also typically a constraint on the number of characters that one can include in a post, therefore providing less contextual information [2]. Besides these processing issues, one of the most prominent challenges with opinion mining from these data sources is differentiating between figurative and literal meanings [1]. Sarcasm detection has become an increasingly popular sub-focus of sentiment analysis for that reason. A lot of text-based data contains words with positive connotations on their own, but when analyzed within the larger context are ironic. This has prompted the recent push in sarcasm detection.

Many machine learning algorithms have been applied to sarcasm detection, with support vector machines achieving the best results in predicting a binary sarcasm label [1]. Other common machine learning algorithms that have been utilized for sarcasm detection include logistic regression, naive Bayes,

and random forests, although, SVMs produced better classification results [1]. The performance of the models utilizing SVMs ranged between 50 and 91 percent, in this study [1], showing there is still room for improvement. Deep learning algorithms, such as convolutional neural networks have also been applied to sarcasm detection and have been used in combination with machine learning models that to provide exceptional results. The utilization of NLP techniques such as part-of-speech (POS) tagging and lexicons have also proved to improve the classification results of standalone machine learning algorithms and those used in combination with CNNs [1]. An SVM-CNN model developed in the same study as the standalone SVM model was able to achieve performance results upward of 97 percent [1]. Another finding from the same study is that it can be beneficial to utilize a two-target variable labeling system where each tweet or piece of text-based data is classified by its sentiment (positive/negative), along with whether it is sarcastic or not (sarcastic/non-sarcastic) [1]. Beyond these text-based models that rely on context clues, there is potential to implement multimodal sentiment analysis approaches to classify unstructured data as sarcastic or not.

Multimodal Sentiment Analysis

Basic sentiment analysis typically involves extracting information from text-based data to infer the sentiment behind such opinions or words in text. Multimodal sentiment analysis goes beyond this and looks to supplement text-based data with audio and visual data. In other words, multimodal sentiment analysis is sentiment analysis with various data sources. Research related to multimodal sentiment analysis has increased due to an increase in multimedia data [4]. For example, multimodal sentiment analysis can be used to analyze the sentiment of someone expressing opinions via a video on social media. Two main applications of multimodal sentiment analysis are analyzing spoken reviews and images that are accompanied by text-based data on social media [4]. Research on this topic is relatively new and other applications will likely arise provided opportunity to explore the topic further. When analyzing spoken opinions via video, typically the three forms of information are independently analyzed and then integrated to conduct multimodal sentiment analysis. The speech modality portion of multimodal sentiment analysis typically looks at para-linguistic features of speech such as tone and pitch of the audio data [4]. Algorithms used to analyze speech data are similar to that of text-based data with many machine learning algorithms. Deep learning algorithms have also been utilized independently or in combination with machine learning algorithms, specifically recurrent neural networks [4]. Visual sentiment analysis often focuses on facial expressions of speakers to associate one's visual emotions with their sentiment towards an entity. Convolutional neural networks are popular in processing and analyzing this form of data for sentiment analysis. Multi-modality combines the features of facial expressions, text-based data, and para-linguistic features [4]. Multimodal sentiment analysis leverages information from these numerous data sources to improve upon unimodal approaches to sentiment analysis.

TABLE I: Overview of the modalities of multimodal sentiment analysis and the typical features and approaches associated with them

Modality	**Example Data**	**Typical Features and Methods**
Text	Product review	Lexicon dictionaries, bag-of-words, and classifiers such as SVM or deep recurrent neural networks
Audio	Speech	Para-linguistic features and classifiers such as SVM or deep recurrent neural networks
Visual	Image	Facial expressions, visual aesthetics, and convolutional neural networks
Multimodal	Video reviews	Fusion of text, para-linguistic features, and facial expressions

Multimodal Sarcasm Detection

Now that we have a better understanding of unimodal text-based sentiment analysis, multimodal sentiment analysis, and sarcasm detection using text-based data, we can begin to explore multimodal approaches to sarcasm detection. Of all the topics discussed within this literature review, this is the least explored area. Multi-modal sarcasm detection is a relatively new idea that combines principles and approaches of all the previously explored topics. Multimodal sarcasm detection looks to improve upon simpler sarcasm detection using text-based data by supplementing text-based data with audio and visual cues to improve classification abilities. We have already touched upon sarcasm detection in text-based modality, so let's focus on reviewing sarcasm detection through the other two modalities involved in multimodal sarcasm detection. Sarcasm detection in speech involves analyzing cues revealed through acoustic patterns [3]. Certain acoustic patterns can be viewed as signs of sarcastic intentions. Some indications of sarcastic tendencies include the rate or intensity of speech [3]. Similar to multimodal sentiment analysis visual cues relevant to sarcasm detection often relate to the facial expressions of the entity expressing such opinion. The contextual information extracted from these other modalities can be combined with textual-based contextual information to provide contextual cues that either support or differ inferences from the text [3]. Initial studies have shown evidence that supports the hypothesis that multimodality provides significant improvement in the detection of sarcasm compared to unimodal approaches [3].

III. The Dataset

Primary Datasets for Multimodal Sentiment Analysis

As multimodal sentiment analysis is a growing area in the field of sentiment analysis, it has its challenges and areas for growth. One of the biggest challenges associated with multimodal sentiment analysis is the limited number of datasets to train and test potential models on. There are two datasets that have become the most prevalent throughout the field: CMU-MOSEI and CMU-MOSI. The CMU-MOSEI dataset is currently the largest dataset utilized for multimodal sentiment analysis and emotion recognition. It contains nearly 24,000 individual data points from over 1000 different speakers. All of the data points are randomly chosen from a wide range of

Utterance: *"Become a drama critic!"*

Emotion: *Joy* **Sentiment:** *Positive*

Text	Audio	Visual
Ambiguous	Joyous tone	Smiling Face

Utterance: *"Great, now he is waving back"*

Emotion: *Disgust* **Sentiment:** *Negative*

Text	Audio	Visual
Positive/Joy	Flat tone	Frown

Fig. 4: Example data points demonstrating the value of multimodal cues. The color indicates the amount each modality contributes to the true sentiment of the statement.

TABLE II: Information contained in dictionary regarding context of each individual clip.

Key	Value
utterance	The text of the target utterance to classify.
speaker	Speaker of the target utterance.
context	List of utterances (in chronological order) preceding the target utterance.
context_speakers	Respective speakers of the context utterances.
sarcasm	Binary label for sarcasm tag.

topics and are all individual speakers (i.e. monologues). Each data point is associated with a sentiment ranging from Positive to Negative, as well as an emotion such as Happiness, Sadness, Anger, or Disgust. While this is an extremely useful dataset for multimodal sentiment analysis in general, we are looking to explore multimodal sentiment analysis applied to sarcasm detection, which is not a feature that is annotated as part of this dataset, and thus this dataset is unsuitable for our work. The CMU-MOSI dataset (the Multimodal Corpus of Sentiment Intensity) is similarly unsuitable for our work as it contains opinion videos annotated with sentiment values ranging from -3 to 3 representing positive and negative sentiment. Since both the CMU-MOSEI and CMU-MOSI datasets do not fit the needs of our work, we had to explore further to discover the MUStARD dataset.

MUStARD Introduction

The dataset that we decided on using for our work is titled the Multimodal Sarcasm Detection Dataset). This dataset consists of a wide variety of audiovisual clips from four different sitcoms: *Friends*, *The Big Bang Theory*, *Golden Girls*, and *Sarcasmaholics Anonymous*. Each audiovisual clip is annotated with a sarcasm label that indicates whether the phrase in a sequence of dialogue (referred to as utterance in the dataset) from the clip is sarcasm or not sarcasm. Each audiovisual clip is represented by two items: the actual video clip from the sitcom, and a dictionary containing the specific text of the data along with context about the clip. Two example clips are displayed in Figure 4 and the information contained in the dictionary is displayed in Table 2.

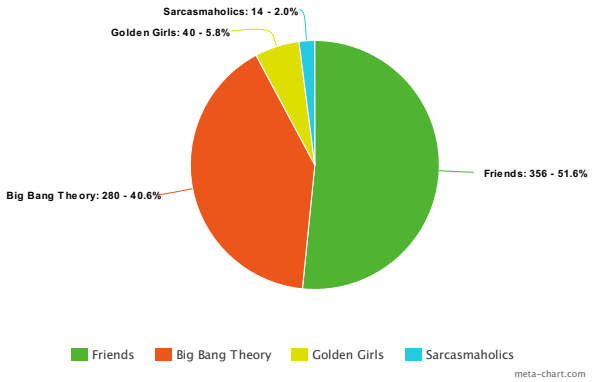

Fig. 5: Distribution of TV shows composing the dataset.

The two example clips in Figure 4 demonstrate the value that including audio and visual information into sarcasm detection can have. In the example said by Monica in *Friends*, the utterance alone can be easily interpreted as having extremely positive sentiment, indicated by the word "Great". But when we look further at the characteristics of the audio of her phrase, we note that there is not actually any joy in her voice and is actually far more deadpan. Finally, visually looking at her facial expression demonstrates that Monica is in fact not happy that someone is waving back at her, giving a strong indication that her original utterance was sarcastic.

Along with the multimodal features that will provide the main building blocks for our model to learn on, there is further annotation associated with each video clip as seen in Table 2. These additional annotations are the speaker of the target utterance, as well as the context of the target utterances (i.e. one to two phrases of dialogue that proceeded the target utterance). As we will see demonstrated later in this paper, both of these additional annotations can allow for improved performance of our sarcasm detection models.

Distribution of Data in MUStARD

The creators of the MUStARD dataset focused the collection of their data from two main data sources: YouTube and MELD, another multimodal emotion recognition dataset that is also based on *Friends*. When selecting clips from *The Golden Girls* and *Sarcasmaholics Anonymous*, the creators focused on almost exclusively on sarcastic examples, while they focused on both sarcastic and non-sarcastic examples from *The Big Bang Theory* and *Friends*. Each of the chosen clips were then manually annotated by two individuals, with a third individual helping to break any split votes. When all the data collection and annotation was said and done, the dataset contained a total of 6,365 videos, only 345 of which were labeled as sarcastic [3]. The creators then decided that they were most interested in a balanced dataset, so they randomly selected an equally sized number of the non-sarcastic clips to make the final size of the dataset 690. We can see in Fig. 5 that more than half of the dataset is comprised of clips from *Friends*, while only a small number of the dataset is comprised of clips from *The Golden Girls* and *Sarcasmaholics Anonymous*

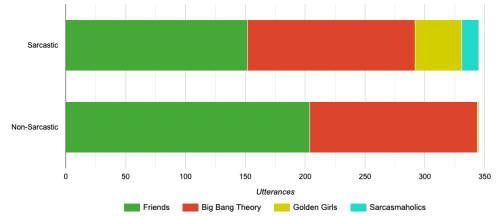

Fig. 6: Distribution of sarcastic and non-sarcastic utterances across the dataset.

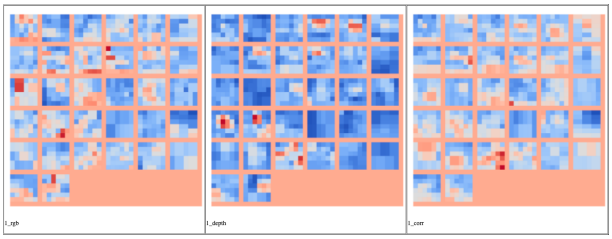

Fig. 7: Example visualization of residual net features extracted from images of a bathtub. The features were normalized to the range [0,1], with cool colors indicating small values and warm colors indication large values [8].

In Figure 5, we can see the distribution between sarcastic and non-sarcastic labels for each of the TV shows in the datset. There are many sarcastic and non-sarcastic examples coming from *Friends* and *The Big Bang Theory*, but only sarcastic examples (aside from one) coming from *The Golden Girls* and *Sarcasmaholics Anonymous*. It will be interesting to see how much our model can learn from these latter two shows given their lack of non-sarcastic examples. Later in this paper, we will explore the performance of our model with a random train-test split between all the samples, as well as by training on all the samples from (pick 3) and testing on (pick 1) to see how the model performs without prior knowledge of the speaker.

IV. Feature Extraction

Obtaining textual features

In order to transform the text of our utterances into usable feature-vectors, we utilized BERT (Bidirectional Encoder Representations from Transformers). BERT is a pre-trained plain language representation model that has usages in many different tasks. BERT was trained on a large amount of unlabeled data from public available sources such as the entirety of Wikipedia, making it an unsupervised model. The model is also deeply bi-directional, meaning it learns information from both the words before and after each token (the specific of interest) [5]. While BERT is often used as a language classification model in fields such as Natural Language Processing, we utilized BERT to generate contextual vector embeddings for each of our utterances. These representations are generated using the hidden layers found in the pre-trained model. We utilized code made publicly available on GitHub by Google Research to extract these vector embeddings with BERT. These vector embeddings were created for both the target utterances and the context phrases for each entry in the dictionary associated with each video clip in the MUStARD dataset, allowing for the consideration of phrase context in the model building step.

Obtaining visual features

In order to extract the visual features from each of the clips, we decided to break down each clip into its individual frames. This would allow us to consistently pre-process and transform each of these individual frames to make sure we are obtaining relevant information from them. The frames were extracted utilizing a bash script that was provided in association with the MUStARD dataset. We implemented a pre-trained residual net model obtained from torchvision, a popular Python package for image transformation and computer vision techniques. The specific residual net model that we utilized was first introduced in "Deep Residual Learning for Image Recognition" where they present a framework that allows for easier training of deeper neural networks. This new learning framework allowed them to develop residual nets with a depth of up to 152 layers [6]. Since we are utilizing a pre-trained model to extract these features, the model is expecting the input images to have the same shape and be normalized in the same way as the data they were trained on. In order to ensure this, we performed a series of transformations including resizing, center-cropping, and normalization as outlined in the torchvision documentation. We also explored greyscaling and random-cropping the frames, but both of these transformations results in slightly worse model performance. Using the residual net model, we extracted our features from the pool_5 activations which look to map to create feature maps from networks based on the RGB, depth, and correlation contents of a given image [7]. We can see an example of these pool_5 feature maps look like for a bathtub in Fig. 7. Obtaining these visual features from each of the frames was by far the most computationally intensive step of the entire process, requiring large amounts of time to fully extract. Because of this, we elected to use a pre-trained residual net model with a depth of 50 layers as opposed to the full depth of 152 layers in the paper. This likely resulted in slightly worse representations of the visual features for our model.

Obtaining audio features

In order to extract audio features from each of the clips, we needed to obtain just the audio data. These audio files were extracted from the original clips using a bash script associated with the MUStARD dataset. In order to extract the audio features we utilized librosa, a popular Python package for music and audio processing. We manipulated the audio through a short-time Fourier transform and were decomposed through filtering nearest neighbors to remove the backgroudn noise in the audio. Then the following features were extracted from each audio file: the mel-frequency cepstral coefficients (MFCC), the first order differences of the MFCCs, the mel-scaled spectrogram, the first order differences of the mel-

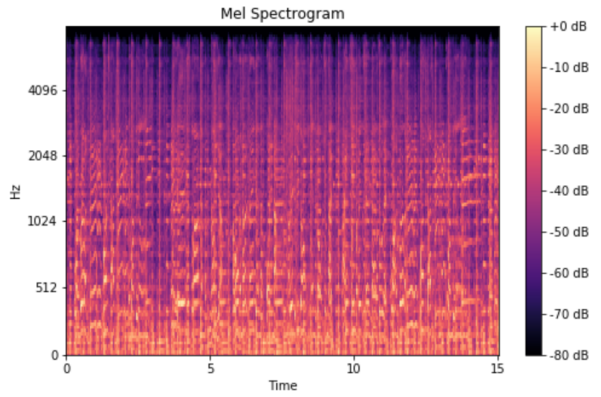

Fig. 8: An sample visualization of a Mel Spectrogram obtained from the librosa package [9].

spectrogram, the spectral centroid, the spectral contrast, the spectral flatness, and the roll-off frequency. Melspectrograms are spectrograms for visualizing the sounds of a given audio file, but instead of looking at the frequency domain, they are explored on the Mel scale. The Mel scale relates to the fact that humans do not perceive frequencies on a linear scale, leading to why we can differentiate between lower frequencies much easier than higher frequencies. The Mel scale is then a transformation of a signal's frequency so that any given distance between two frequencies on the Mel scale is always perceived to be the same to humans [9]. We can see a visualization of the information contained in a melspectrogram in Fig. 8. All of these spectral features that we extracted from the audio are focused on identifying aspects of speech such as notes, pitch, rhythm, and melody to give an indication of sarcasm.

V. Multimodal Sarcasm Detection

Model Selection

To explore the role that multi-modal features can have in performing sarcasm detection, we begin by analyzing the performance of several untuned classification models from scikit-learn. For this analysis, we will look at the speaker dependent case (data points from each TV show are found in both the training and test sets). We will set out baseline model as random binary predictions between sarcastic and non-sarcastic for each data point. As we are analyzing a speaker dependent case, we will perform 10-fold cross-validation to guarantee consistency between the results of the different models. The cross-validation model will ensure that the ratio of sarcastic to non-sarcastic samples in each fold will be consistent. To evaluate the quality of our model, we will use scikit-learn's builtin classification report function which will provide us with precision, recall, f1-score, and support. We then aggregate the results of the classification report across each fold of the cross-validation to determine each models overall performance. We will analyze the performance of the following models: Logistic Regression, Support Vector Machine, K-Nearest Neighbors, Gaussian Process, Random Forest, Multi-Layer Perceptron, XGBoost, Gaussian Naive Bayes, and Quadratic Discriminant Analysis. We can see the performance of each of our models in Table 3.

TABLE III: Performance of Different Classification Models

Model	Precision	Recall	F1-Score
Random	0.473	0.473	0.473
Logistic Regression	0.669	0.667	0.665
SVM Classifier	0.692	0.690	0.690
KNN Classifier	0.599	0.596	0.591
GP Classifier	0.552	0.507	0.351
RF Classifier	0.654	0.648	0.645
MLP Classifier	0.683	0.680	0.679
XGBoost	**0.703**	**0.702**	**0.701**
Gaussian NB	0.645	0.578	0.521
QDA	0.495	0.495	0.471

These models were trained using features from all three modalities, and thankfully all of the models were able to outperform a random prediction, immediately giving us evidence that the features we extracted in the previous section have at least some predictive value in determining whether a given utterance is sarcastic or non-sarcastic. Our highest performing models were eXtreme Gradient Boosting, Support Vector Machines, Multi-Layer Perceptron. We were specifically surprised that the MLP model was on of our three best performing models because feedforward neural networks like MLP classifiers typically need a large amount of data to determine the hidden weights for each of their layers, and we were not applying the model to a particularly large dataset. On the other hand, we were not surprised that the SVM classifier performed well because it tends to perform well with smaller amounts of data. While both SVM and MLP classifiers performed well, the highest performing model was an XGBoost classifier. XGBoost is a machine learning model that combines decision trees with ensemble learning that looks to minimize model loss and complexity. XGBoost performs well for a wide range of machine learning tasks so we were not surprised that it performed well for our problem.

Feature Importance

Now that we have decided upon SVM, we explored the value that each of our modalities (text, audio, and video) have on our model performance. To start performing this exploration, we followed a similar procedure as our model selection process, namely performing 10-fold cross-validation and using a classification report to evaluate the performance of each model. In order to get a better idea of each modalities contribution to sarcasm detection, we looked at the performance of unimodal model (a model trained on only one of our modalities), bimodal model (a model trained on only two of our modalities), and the full multimodal model (a model trained on all three modalities). This analysis will provide us with the results of for a speaker dependent case, but we are also interested in seeing the value of each feature when our model has never seen the speaker before. To perform this, we will take two different approaches. The first will be to include all the clips from Friends, The Big Bang Theory, and Sarcasmaholics Anonymous, but withhold clips from the Golden Girls. The second will be to include clips from all TV shows, but without all clips spoken by (pick a person in one

TABLE IV: Classification Performance with Various Features (Speaker Dependent)

Features	Precision	Recall	F1-Score
Random	0.473	0.473	0.473
Text	0.640	0.638	0.637
Audio	0.681	0.679	0.678
Video	0.667	0.664	0.662
Text + Audio	0.701	0.699	0.698
Text + Video	**0.726**	**0.725**	**0.724**
Audio + Video	0.700	0.698	0.698
Text + Audio + Video	0.704	0.702	0.701

TABLE V: Classification Performance with Various Features (*The Big Bang Theory* withheld)

Features	Precision	Recall	F1-Score
Random	0.507	0.507	0.507
Text	0.552	0.543	**0.522**
Audio	0.621	0.564	0.506
Video	0.505	0.504	0.475
Text + Audio	**0.660**	**0.579**	0.517
Text + Video	0.593	0.554	0.501
Audio + Video	0.563	0.536	0.480
Text + Audio + Video	0.530	0.514	0.440

of the shows with a lot of clips probably Sheldon). The results of these three approaches can be found in Tables 4, 5, and 6.

Table 4 demonstrates the performance of a speaker dependent classification model trained on various combinations of the features extracted in the previous section of the report. When it comes to utilizing only one of the features, we found that the audio features provide the strongest performance while textual features provide the weakest performance. While we were not expecting the textual features to have the worst performance out of the three, this provides even more evidence for the basis of the motivation behind our work to improve sarcasm detection methods by utilizing features beyond text. The highest unimodal performance established by audio is improved with the addition of either text or video in a bimodal model, but the highest performing bimodal model is trained on textual and visual features combines. Surprisingly, this bimodal model outperformed the multimodal model. From the beginning of our analysis, we expected the multimodal model trained on textual, audio, and visual features to perform the strongest out of any combination, but this was not the case. What makes this even more interesting is that audio is the strongest modality when used on its own, but is not part of the highest performing model. This result indicates that some more elaborate fusion between modalities than just concatenation of features is necessary for high performing sarcasm detection.

Table 5 demonstrates the performance of a show independent classification model trained on various combinations of modalities. As we had expected, the performance of the model is worse than the speaker-dependent setup. Again audio proves to be the most important unimodal feature with visual features performing the worse. In this case, the bimodal model trained on textual and audio feautres performs the strongest of all models, and the full multimodal model has the second-worst performance. We believe that the textual and audio features were much more impactful in the show-independent setting because there is more consistency between topics of

TABLE VI: Classification Performance with Various Features (Chandler withheld)

Features	Precision	Recall	F1-Score
Random	0.452	0.435	0.414
Text	**0.596**	**0.617**	**0.495**
Audio	0.579	0.595	0.477
Video	0.562	0.581	0.489
Text + Audio	0.567	0.582	0.470
Text + Video	0.584	0.604	0.488
Audio + Video	0.526	0.531	0.421
Text + Audio + Video	0.572	0.579	0.442

TABLE VII: Classification Performance with Context and Speaker

Features	Precision	Recall	F1-Score
Audio	0.681	0.679	0.678
+ Speaker	0.678	0.677	0.676
+ Context	0.669	0.667	0.665
+ Context & Speaker	**0.684**	**0.681**	**0.681**
Text + Video	**0.726**	**0.725**	**0.724**
+ Speaker	0.725	0.723	0.722
+ Context	0.717	0.715	0.714
+ Context & Speaker	0.723	0.717	0.716
Text + Audio + Video	0.704	0.702	0.701
+ Speaker	**0.719**	**0.715**	**0.714**
+ Context	0.693	0.690	0.689
+ Context & Speaker	0.718	**0.715**	**0.714**

conversation and the tone and pitch of speaker's voices when being sarcastic between the shows than with visual features. Since our visual features were extracted from still frames and not the entire video, they likely struggle to capture some of the facial expression and body-cues that indicate sarcasm. Thus when a completely new individual and environment is introduced, the model may not know how to interpret as well.

Table 6 demonstrates the performance of a speaker independent classification model trained on various combinations of modalities. As opposed to the setting explore in Table 5, this version of the model was trained on clips from all of the TV shows, but all of the utterances spoken by Chandler (from *Friends*) were withheld. Surprising to us, this version of the model performs even worse than the show independent model. The highest performing model turns out to be the unimodal model trained only on textual features, which was not something we were expecting. These results indicate that the techniques present in this paper are not sufficient for predicting sarcasm for an unseen speaker and that there is much more work that can be done in this area.

Speaker and Context

Along with the text, audio, and visual features, the data is also annotated with the context surrounding the specific statement of interest, as well as the individual that made the given statement. Utilizing the models that performed the best from Table 4, we will now look to see if the performance of these models can be improved by including these contextual and speaker features. The results of including these features can be seen in Table 7, applied to a unimodal model with audio features, a bimodal model with textual and visual features, and a trimodal model with textual, audio, and visual features. As we can see by the bold-faced font, both the unimodal and tri-

modal models have a performance increase form the inclusion of both contextual and speaker features. Interestingly enough, the performance of the strongest overall model (the bimodal model) does not improve with the inclusion of contextual and speaker features.

Hyperparameter Tuning

In order to further improve the performance of our model, we looked to tune the hyperparameters of XGBoost. The three specific hyperparameters that we looked to tune were the learning rate (eta), the max tree depth, and the minimum loss reduction (gamma). To tune these hyperparameters we performed 3-fold cross-validation to see which sets of values results in the strongest performance. The results of the tuning step are found in Table 8.

From the results in Table 8, we see that the best values for our hyperparameters of learning rate, maximum depth, and minimum loss reduction are 0.3, 3, and 0 respectively. Performing the same 10-fold cross-validation as we did in the model exploration phase, we again evaluated the performance of our final classification model for sarcasm detection. We obtained final classification metrics of 0.721, 0.716, and 0.714 for weighted precision, weighted recall, and weighted f1-score.

Failure Cases

In Figure 9 we see two interesting cases that our model classified incorrectly. The first case is an utterance from Leonard in *The Big Bang Theory* that was sarcastic, but our model classified as non-sarcastic. There are multiple different aspects of this clip that made it more likely to be misclassified. First, Leonard cannot actually be seen in the clip, as he is on the other side of the door, so the visual features cannot provide any

TABLE VIII: Hyperparameter Tuning of XGBoost

eta	max_depth	gamma	Precision	Recall	F1-Score
0.1	12	0	0.68	0.678	0.677
0.1	12	0.005	0.697	0.694	0.693
0.1	12	0.01	0.708	0.706	0.705
0.3	12	0	0.699	0.697	0.696
0.3	12	0.005	0.693	0.691	0.69
0.3	12	0.01	0.693	0.691	0.69
0.5	12	0	0.698	0.694	0.693
0.5	12	0.005	0.695	0.691	0.69
0.5	12	0.01	0.708	0.704	0.703
0.1	6	0	0.68	0.678	0.677
0.1	6	0.005	0.683	0.68	0.678
0.1	6	0.01	0.689	0.684	0.682
0.3	6	0	0.687	0.686	0.685
0.3	6	0.005	0.675	0.674	0.673
0.3	6	0.01	0.688	0.687	0.686
0.5	6	0	0.684	0.684	0.684
0.5	6	0.005	0.696	0.696	0.696
0.5	6	0.01	0.673	0.672	0.672
0.1	3	0	0.696	0.693	0.691
0.1	3	0.005	0.696	0.693	0.691
0.1	3	0.01	0.696	0.693	0.691
0.3	3	0	**0.711**	**0.709**	**0.708**
0.3	3	0.005	0.705	0.703	0.702
0.3	3	0.01	0.705	0.703	0.702
0.5	3	0	0.7	0.699	0.698
0.5	3	0.005	0.7	0.699	0.698
0.5	3	0.01	0.7	0.699	0.698

Fig. 9: Misclassified utterances.

information about the sarcasm in the utterance. Additionally, Leonard's voice stays almost completely monotone throughout the phrase, providing no indication of sarcasm through a change in tone or pitch. The second case is an utterance from Chandler in *Friends* that was not sarcastic, but our model classified as sarcastic. This utterance was likely misclassified because of the change in the tone of Chandler's voice as he is making a joke. This indicates that the model may be picking up on humor and using that as an indicator of sarcasm.

VI. Conclusions and Future Work

Overall, our initial exploration of applying machine learning models to sarcasm detection shows promising results. We immediately found that only textual data does not provide sufficient information for performing sarcasm detection in our analysis of unimodal models. We also found the performance of our model continued to increase as we increased the modality, with a bimodal model utilizing textual and visual data outperforming the best unimodal model, and a trimodal model utilizing textual, audio, and visual data outperforming all the unimodal and bimodal models. We also found that including the context and the speaker as features generally increase the performance of our models. Finally, upon an exploration of various types of machine learning models, we found support-vector-machines to be the highest performing model.

Our paper and methodology could be improved upon in numerous ways. For one, there is still significant area left for improvement related to the feature extraction of textual features. A generalized autoregressive pretraining method called XLNet could be used as a substitute to BERT [10]. Due to its autoregressive formulation, XLNet has been proven to overcome shortcomings of the BERT model and outperform BERT on a multitude of tasks including sentiment analysis [10]. Another improvement that could be made to increase the performance of our model includes further analyzing which spectral components provide the most value for this task. The third and final area of improvement we identified related to feature extraction, is extracting information related

to individuals movements as opposed to the frame-by-frame elements.

The multimodal fusion our approach utilizes is that of data level fusion. There is potential for improvement in our model performance by using late fusion or decision level fusion instead. Decision level differentiates from data level fusion by dealing with errors from different models independently.

In terms of future work, we can move beyond the domain of sitcom data and investigate the model performance on other sources of sarcastic data where the sarcasm is less exaggerated. Other exploration areas include exploring whether the model can recognize cultural differences in sentiment or differences related to the time period of the data.

REFERENCES

[1] Sarsam, S. M., Al-Samarraie, H., Alzahrani, A. I., &; Wright, B. (2020). SARCASM detection using machine learning algorithms in Twitter: A systematic review. International Journal of Market Research, 62(5), 578–598. https://doi.org/10.1177/1470785320921779

[2] Poria, S., Hazarika, D., Majumder, N., &; Mihalcea, R. (2020). Beneath the tip of the iceberg: Current challenges and New Directions in sentiment analysis research. IEEE Transactions on Affective Computing. https://doi.org/10.1109/taffc.2020.3038167

[3] Castro, S., Hazarika, D., Pérez-Rosas, V., Zimmermann, R., Mihalcea, R., &; Poria, S. (2019). Towards Multimodal Sarcasm Detection (An _Obviously_ Perfect paper). Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. https://doi.org/10.18653/v1/p19-1455

[4] Soleymani, M., Garcia, D., Jou, B., Schuller, B., Chang, S.-F., &; Pantic, M. (2017). A survey of multimodal sentiment analysis. Image and Vision Computing. https://doi.org/https://doi.org/10.1016/j.imavis.2017.08.003

[5] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[6] He, K., Zhang, X., Ren, S., & Sun, J. (205). Deep Residual Learning for Image Recognition. https://doi.org/10.48550/arXiv.1512.03385.

[7] Xu, X., Li, Y., Wu, G., & Luo, J. Multi-modal Deep Feature Learning for RGB-D Object Detection. https://doi.org/10.1016/j.patcog.2017.07.026.

[8] Nanjing Univeristy, Department of Computer Science and Technology Media Computing Research Group (2015). pool5 features. http://mcg.nju.edu.cn/dataset/pool5/

[9] Leland, R. Understanding the Mel Spectrogram. https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53

[10] Yang, Z., Dai, Z., Carbonell, J., Salakhutdinov, R., amp; Le, Q. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. Computation and Language, 2. https://doi.org/https://doi.org/10.48550/arXiv.1906.08237