Augmenting pre-trained language models with audio feature embedding for argumentation mining in political debates

Rafael Mestre[†], **Stuart E. Middleton**, **Matt Ryan**, **Masood Gheasi**, **Timothy J. Norman** and **Jiatong Zhu**

University of Southampton
[†]r.mestre@soton.ac.uk

Abstract

The integration of multimodality in natural language processing (NLP) tasks seeks to exploit the complementary information contained in two or more modalities, such as text, audio and video. This paper investigates the integration of often under-researched audio features with text, using the task of argumentation mining (AM) as a case study. We take a previously reported dataset and present an audio-enhanced version (the Multimodal USElecDeb60To16 dataset). We report the performance of two text models based on BERT and GloVe embeddings, one audio model (based on CNN and Bi-LSTM) and multimodal combinations, on a dataset of 28,850 utterances. The results show that multimodal models do not outperform text-based models when using the full dataset. However, we show that audio features add value in fully supervised scenarios with limited data. We find that when data is scarce (e.g. with 10% of the original dataset) multimodal models yield improved performance, whereas text models based on BERT considerably decrease performance. Finally, we conduct a study with artificially generated voices and an ablation study to investigate the importance of different audio features in the audio models.

1 Introduction

In recent years, there has been an increasing interest in multimodal classification, which refers to the task of automatically classifying an input based on multiple modalities or sources of information, such as text, images and audio (Baltrušaitis et al., 2018). Multimodal approaches are beneficial as they can reduce the subjectivity of classification with a single modality and improve the accuracy of the overall classification. However, finding the best representations (especially those that work well with other modalities), aligning and fusing them, and getting the models to co-learn are difficult challenges to overcome (Morency and Baltrušaitis, 2017). A large body of literature has focused on the combination of image and text for applications like emotion recognition (Illendula and Sheth, 2019), fake news detection (Nakamura et al., 2020), image classification (Guillaumin et al., 2010), or document image classification (Jain and Wigington, 2019). Much less attention has been paid to combining audio with text.

Audio can convey a variety of information about the pitch or intonation of the speaker that can indicate variance in emotional state as well as better identify modes of communication like sarcasm that have been difficult for models to detect. The integration of audio has successfully improved classification tasks like multimodal sentiment analysis and emotion recognition when compared with classic NLP models (Yao et al., 2020; Ho et al., 2020). In this paper, we focus on a less explored NLP area in terms of multimodality: argumentation mining (AM). AM is the computational study of arguments to develop models that can automatically identify, extract, and represent arguments in text or other forms of digital communication such as audio or video. AM has traditionally focused on textual data such as news articles, blog posts, and online comments, but the advantages of using audio to detect arguments have not been extensively explored.

In this work, we expand on an existing AM dataset of US political debates (USElecDeb60To16 by Haddadan et al. 2019) by including audio. We test the performance of several multimodal AM models in different variations of the same dataset, e.g., after balancing the labels and with fractional datasets. Our contribution is three-fold: i) a new fully aligned audio dataset, expanding on an existing AM dataset (Section 3), adding balanced and fractional subsets for researchers to experiment with; ii) original multimodal benchmarking results for this dataset highlighting where audio feature embeddings add most value compared to text-only models (Section 4); iii) analysis of audio features importance, including performance comparison of

Findings of the Association for Computational Linguistics: EACL 2023, pages 274–288
May 2-6, 2023 ©2023 Association for Computational Linguistics

human and computerized voices (Section 5) and an ablation study (Section 6).

2 Related work

Multimodal approaches including audio have been mostly used for sentiment analysis or emotion recognition (Yang et al., 2022; Cai et al., 2019), often using the IEMOCAP dataset, one of the oldest datasets that contains 12 hours of dialogue recordings with emotion labels in text, audio and video format (Busso et al., 2008). In recent years newer datasets have been released, such as the SAVEE (Jackson and Haq, 2014) and RAVDESS databases (Livingstone and Russo, 2018), the MELD dataset (Poria et al., 2018), and the CNU-MOSEI dataset (Zadeh et al., 2016). Generally, audio-textual multimodal approaches contain separate pipelines for audio and text features, sometimes connected through attention layers. For instance, Cai *et al.* (2019) combined GloVe embeddings in a bidirectional long-short term memory (Bi-LSTM) array for text with a combination of a convolutional neural network (CNN) and a Bi-LSTM array for the audio. Likewise, Yoon et al. (2018) used GloVe embeddings and recurrent neural networks (RNN) for both audio and text, reaching accuracies of 71.8 % with the IEMOCAP dataset. Atmaja and Akagi (2020) used either LSTM or CNN (not at the same time) for the acoustic pipeline and LSTM with FastText and GloVe embeddings. Ho et al. (2020) used a multi-level multi-head fusion attention using bidirectional encoder representations (BERT) for the text representations, achieving improved accuracies in three different datasets.

Audio features in this domain have been generally embedded using low level descriptors (LLDs), such as mel-frequency cepstral coefficients (MFCCs) (Atmaja and Akagi, 2020; Ho et al., 2020). MFCCs are computed from the mel spectrogram of the audio signal by performing a discrete cosine transform (DCT) of its log to reduce its dimensionality in a way that is highly related to the raw signal, but approximating the human auditory system and often yielding higher classification performance (Singh et al., 2021). As LLDs do not contain global information about the utterance, high-level statistical functions (HSFs), such as mean, kurtosis and quadratic error, among many others, can also be used. Yao et al. (2020) compared the performance in speech emotion recognition of a HSF classifier based on a deep neural network (DNN), a LLS classifier based on a recurrent neural network (RNN) and a raw-signal mel-spectrogram classifier based on a CNN, finding similar performance between the HSF and LLD models, and a slightly lower performance for the model using the raw signal, showing the benefits of the low level representations. The use of RNN with LLDs has been shown to offer benefits by considering the temporal dimension of an utterance (Xie et al., 2019), but several researchers have started to use both CNNs and RNNs in combination to learn both temporal and local features in the frequency domain (Zhao et al., 2019; Singh et al., 2021; Yao et al., 2020). Whereas MFCCs are the feature of choice for the great majority of applications, the list of remaining LDDs are virtually endless, including the zero crossing rate, chroma vector, entropy of energy, Hammarberg index, spectral slope, harmonic difference, among many others (Atmaja and Akagi, 2020). While some efforts have been made towards standardization of audio features (Eyben et al., 2016), the choice is generally pragmatic and depends on the package used by the researcher, with openSMILE toolkit (Eyben et al., 2013), Librosa (McFee et al., 2015) and PyAudioAnalysis (Giannakopoulos, 2015) being those most commonly chosen ones.

On the other hand, AM research has focused on a diverse set of applications using the text modality alone, from online interactions (Ghosh et al., 2014) and tweets (Alsinet et al., 2019) to argumentative essays (Stab and Gurevych, 2014) and political debates (Lawrence and Reed, 2017; Visser et al., 2021). Regarding multimodal AM, Lippi and Torroni (2016) presented a first step towards the use of audio features from speech to improve argument detection. In this paper, they used raw input signals, which were passed through a speech recognition API to obtain the text. Then they used bag of words and bi-grams together with discrete HSF features from MFCCs, namely minimum, maximum, average and standard deviation, to train a support vector machine in an argument classification task. The results were positive towards the addition of audio, although the performance was modest due to the small size of the dataset and the limitations of the text and audio representations. The only other work that considered multimodal aspects used the M-arg dataset (Mestre et al., 2021). There, the authors analyzed argumentative relations in the 2016 US presidential debates using text and audio, building an argumentation mining pipeline based on BERT embeddings for text and a combination of a Bi-LSTM and a CNN for the audio. Although the dataset, annotated for "support" and "attack" between sentences, was rather small and heavily unbalanced towards the "neither" class, the authors reported a slight improvement when considering audio and text together in a multimodal model. Surprisingly, audio features alone showed a better performance than the text-only model based on BERT encodings, suggesting that in small datasets, when the performance of BERT-based models suffers, audio features might provide a handy supplement to classify arguments. The effect of specific audio features on performance was not assessed.

Here, we build upon a previous dataset (USElecDeb60To16) presented by Haddadan et al. (2019), which contained English transcripts of the US presidential debates from 1960 to 2016 labelled with more than 29k annotations of argument components and their boundaries. We used the original videos from the debates to obtain aligned timestamps at the sentence level following the work of Mestre et al. (2021), thus enabling the task of multimodal AM with a total of 28,850 aligned and annotated sentences. Concurrently to the submission of our work, Mancini et al. (2022) also presented and released a multimodal dataset, using the same videos and alignment process, with 26,791 sentences. Both datasets are complementary, although our dataset is slightly larger as we did not drop any of the debate videos (see next section). Mancini et al. (2022) compare datasets and architectures from the two previously mentioned works by Lippi and Torroni (2016) and Mestre et al. (2021), as well as their new dataset, finding generally positive results to the addition of audio. In our work, we report an audio feature analysis, as well as the impact of using computerized voices, and we investigate the benefit of multimodal models on both balanced and fractional small data subsets.

3 Methodology

3.1 Dataset construction

In their USElecDeb60To16 dataset, Haddadan et al. (2019) reported the performance of several models for argument classification, with the highest weighted F-score of 0.673 for the argumentative component classification (ACC) of premise, claim and other. They also collapsed all the premise/claim annotations into one single label, "argument", and attempted argumentative sentence detection (ASD), with a weighted F-score of 0.843 using an LSTM array. We used this dataset in its collapsed version (argument/other) for ASD to assess whether the addition of audio could improve the reported performance (F-score of 0.843) and to simplify the task using only 2 classes as a first step to studying the potential of multimodal AM. For this, we needed to add the audio of the debates with sentence-level timestamps.

Videos from each debate were downloaded from the YouTube channel of the Commission for Presidential Debates.[1] Before starting the audio alignment process, we fixed a small number of inconsistencies in the dataset resulting from errors in the original transcripts. Some were simple, like sentences lacking a space between periods, which made sentence tokenization algorithms fail. In a couple of debates, full paragraphs were missing from the transcript, possibly due to an error in the original web scraping algorithm by Haddadan et al. (2019). Older debates also had serious transcription issues in the original source, such as full sentences or paragraphs missing or speeches being repeated twice in the transcript. Regarding videos, older ones also had issues, such as debate 5 (the first Carter-Ford Debate in 1976), in which the audio was lost during live transmission, and commentators, not presidential candidates, spoke for almost half an hour. This was not reflected in the transcript, and we had to manually edit the video to match the transcript. For two debates (the first and second Clinton-Bush-Perot debates of 1992), the Commission decided to split the transcript into two parts, even though the debates occurred uninterrupted. Therefore, we split the videos in two to match the transcript. Others had cuts or repeated segments that lasted from a few seconds to several minutes, and we were forced to adapt the original dataset to reflect these changes. Whereas preceding researchers Mancini et al. (2022) were forced to remove full or part of the debates from their multimodal dataset to account for these issues, we rigorously edited the USElecDeb60To16 dataset and videos to reduce unsystematic data loss error, such that we could provide an enhanced comprehensive dataset to researchers for further investigation. We want to highlight that this error reduction does not cast any doubt on the quality and substantive find-

[1] https://www.debates.org/voter-education/debate-transcripts/

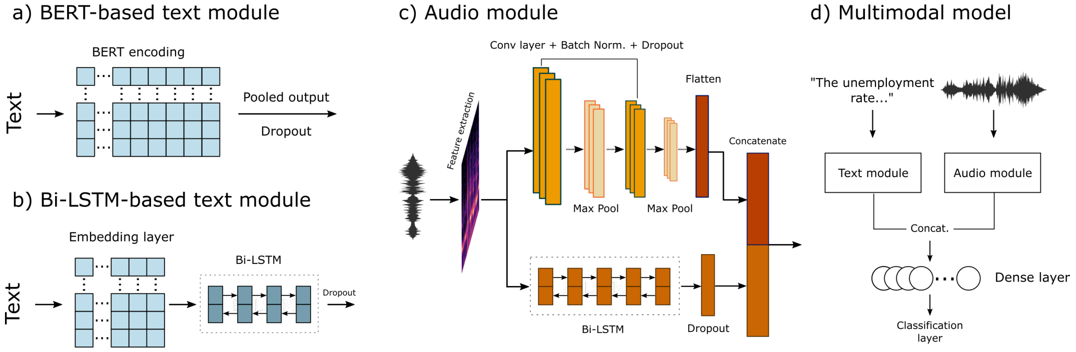

Figure 1: Model architectures used in this paper. a-c) Generic architectures of the text and audio modules. d) Multimodal model that combines text and audio modules.

ings of that preceding work, as it mostly relates to sentence omission.

We aligned the transcripts with the audio using the Aeneas Forced Alignment (v. 1.7.3) tool as proposed by Mestre et al. (2021). Two researchers manually checked every debate for major misalignment (and fixed them) until we obtained an almost perfectly aligned text. After alignment, our dataset contained 28,850 labelled sentences (76.15% of them arguments), with timestamps indicating start and ending times in the audio file. We present the extended dataset we call Multimodal USElecDeb60To16, with the collapsed original annotations, alongside timestamps to match the audio, instructions for obtaining the videos and scripts to extract the audio features, in our GitHub repository page (Mestre et al., 2023).[2]

3.2 Model architectures

We used the architectures proposed by Mestre et al. (2021), which showed the potential of multimodal argumentation mining in our dataset (Figure 1). We considered two text modules based on GloVe and BERT (Devlin et al., 2018). The former used Wikipedia-trained 200-dimensional GloVe embeddings, whose maximum length was given by its 99th percentile to eliminate very long sentences. They were passed through a Bi-LSTM, followed by a dropout layer, a dense layer and an output layer with softmax activation. The latter module consisted of a BERT pre-processor[3] and a BERT encoder with L=12 hidden layers, a size of H=768 and A=12 attention heads.[4] Its pooled output was also followed by a dropout layer and a dense layer.

The audio module was inspired by Cai et al. (2019). For each utterance in audio form, the Python library Librosa was used for audio feature extraction (McFee et al., 2015). We extracted the following LLD features: MFCCs (Klapuri and Davy, 2006), spectral centroids (Klapuri and Davy, 2006), spectral bandwidth (Klapuri and Davy, 2006), spectral roll-off (McFee et al., 2015), spectral contrast (Jiang et al., 2002a), and a 12-bit chroma vector (McFee et al., 2015). Motivation for selection of features and evaluation is further described in Section 6. For each sentence, the features were concatenated to form a tensor of $(45, T)$, where T is the duration of the utterance. All utterances were padded with zeros to have the same length T_{max}, which was defined by the 99th percentile duration of all utterances. Each utterance was passed in parallel through a CNN and a Bi-LSTM to find both local and temporal features. The CNN consisted of two convolutional layers, two maxpool layers and batch normalization layers. Outputs from both modules were flattened, concatenated and passed through dropout and dense layers.

The multimodal model was a combination of the text and audio modules in which the inputs were the text string and its corresponding audio, each passed in parallel. We considered two multimodal models: one with a BERT text module and another one with a Bi-LSTM text module.

3.3 Hyperparameter tuning

We developed a robust methodological framework to tune the hyperparameters for each model. Model training was performed in a High Performance Computing (HPC) cluster in dedicated GPUs (with either nodes of 4 GTX1080 Ti GPUs or nodes of 2 Nvidia Volta V100 GPUs). The hyperparame-

[2] https://github.com/rafamestre/Multimodal-USElecDeb60To16.

[3] bert_en_uncased_preprocess v.3

[4] bert_en_uncased_L-12_H-768_A-12 v.4

Model	Class	Original dataset ($N = 28,850$)			Balanced dataset ($N = 13,758$)		
		Precision	**Recall**	F_1	**Precision**	**Recall**	F_1
Text Bi-LSTM	Argument	0.844 ± 0.005	0.950 ± 0.011	0.893 ± 0.004	0.707 ± 0.011	0.789 ± 0.037	0.745 ± 0.012
	Other	0.727 ± 0.028	0.429 ± 0.034	0.539 ± 0.023	0.761 ± 0.020	0.669 ± 0.039	0.711 ± 0.017
	Wt. average	0.816 ± 0.006	0.827 ± 0.005	0.810 ± 0.006	0.734 ± 0.007	0.730 ± 0.007	0.729 ± 0.007
	Macro av.	0.785 ± 0.013	0.690 ± 0.013	0.716 ± 0.012	0.734 ± 0.007	0.729 ± 0.007	0.728 ± 0.007
Text BERT	Argument	0.854 ± 0.004	0.951 ± 0.006	0.900 ± 0.002	0.714 ± 0.013	0.839 ± 0.025	0.771 ± 0.011
	Other	0.758 ± 0.018	0.487 ± 0.018	0.593 ± 0.012	0.813 ± 0.017	0.674 ± 0.024	0.737 ± 0.010
	Wt. average	0.831 ± 0.004	0.839 ± 0.003	0.826 ± 0.004	0.764 ± 0.007	0.755 ± 0.006	0.754 ± 0.006
	Macro av.	**0.806 ± 0.008**	**0.719 ± 0.007**	**0.746 ± 0.006**	**0.763 ± 0.007**	**0.757 ± 0.005**	**0.754 ± 0.006**
Audio	Argument	0.785 ± 0.011	0.973 ± 0.028	0.869 ± 0.005	0.628 ± 0.032	0.517 ± 0.279	0.521 ± 0.204
	Other	0.654 ± 0.081	0.135 ± 0.080	0.211 ± 0.089	0.603 ± 0.063	0.670 ± 0.211	0.612 ± 0.064
	Wt. average	0.754 ± 0.011	0.775 ± 0.003	0.714 ± 0.017	0.615 ± 0.017	0.595 ± 0.036	0.567 ± 0.078
	Macro av.	0.720 ± 0.035	0.554 ± 0.026	0.540 ± 0.042	0.615 ± 0.016	0.593 ± 0.034	0.566 ± 0.080
Multimodal (Bi-LSTM +Audio)	Argument	0.879 ± 0.031	0.672 ± 0.255	0.733 ± 0.184	0.765 ± 0.054	0.548 ± 0.259	0.593 ± 0.230
	Other	0.451 ± 0.127	0.681 ± 0.173	0.515 ± 0.053	0.661 ± 0.087	0.812 ± 0.104	0.719 ± 0.021
	Wt. average	0.776 ± 0.016	0.674 ± 0.153	0.680 ± 0.152	0.713 ± 0.019	0.679 ± 0.079	0.656 ± 0.126
	Macro av.	0.665 ± 0.051	0.677 ± 0.046	0.624 ± 0.117	0.713 ± 0.019	0.680 ± 0.078	0.656 ± 0.126
Multimodal (BERT +Audio)	Argument	0.851 ± 0.007	0.940 ± 0.006	0.893 ± 0.006	0.730 ± 0.031	0.773 ± 0.057	0.749 ± 0.014
	Other	0.723 ± 0.016	0.487 ± 0.020	0.581 ± 0.016	0.762 ± 0.033	0.712 ± 0.059	0.734 ± 0.017
	Wt. average	0.820 ± 0.009	0.830 ± 0.008	0.818 ± 0.009	0.746 ± 0.004	0.742 ± 0.005	0.741 ± 0.005
	Macro av.	0.787 ± 0.011	0.713 ± 0.010	0.737 ± 0.010	0.746 ± 0.005	0.743 ± 0.004	0.741 ± 0.005

Table 1: Models' performance for original and balanced datasets. Errors indicate standard deviation after 5 replicates.

ter training was assisted by the Python package Ray[tune] (Liaw et al., 2018), which allows distributed parallel hyperparameter tuning with different search strategies and schedulers. We defined our hyperparameter search as shown in the appendix (Table A1), including training parameters like the learning rate and batch size, and also architecture-dependent parameters like the kernel size of the CNN or whether the text embeddings should be retrained or not. To search over the defined hyperparameter space, we used the Tree-structured Parzen Estimator algorithm (Bergstra et al., 2011), which considers the performance of previous iterations of the search to choose the next hyperparameters to test, implemented in the HyperOpt library for parallel optimization (Bergstra et al., 2013). The training was implemented in TensorFlow and we included Keras callbacks after each epoch of training to update the hyperparameter search algorithm. We implemented an early stopping scheduler algorithm that monitored validation loss at each epoch, stopping the training before overfitting, with a minimum change of 1e-4 and a patience of 3 epochs. We considered implementing other schedulers like Asynchronous Successive Halving Algorithm (ASHA), which stops unpromising trials if their performance is worse than that of previous trials (Li et al., 2018), but we discovered in our experiments that this algorithm tends to penalize slow learning models, which sometimes ended up giving the best results. We sampled 50 times the search space, with validation split of 20% and test split of 20%, and the best hyperparameters, used in the remaining sections, are reported in the appendix (Table A1), as well as average runtimes and number of parameters. With our dataset, we provide full details of the training results, with confusion matrices, training history, validation metrics plots, etc.

4 Models' performance

4.1 Full original dataset

Table 1 shows the performance of each one of the models after training with the original dataset ($N = 28,850$) and optimized parameters. The text-only models, particularly the BERT model, perform best in terms of both macro and weighted F-scores, reaching a weighted average of 0.826 (macro average of 0.746), comparable to the weighted average reported by Haddadan et al. (2019) of 0.843 (or macro average of 0.730) using a LSTM network. As in that paper, the precision and recall of the "other" class is low, but classification of the "argument" class performs much better, with a high recall in both the BERT and Bi-LSTM models. The audio-only model yields a rather low macro averaged F-score of 0.540, as it relies on over-classifying the argument class. The BERT-based multimodal model performs significantly better than the audio-only model and similarly to the text models, with a macro average of 0.737. The Bi-LSTM-based multimodal model performs better than the audio-only model in terms of macro

Model	**Class**	10% original dataset ($N = 2,885$) **Precision**	**Recall**	F_1	10% balanced dataset ($N = 1,376$) **Precision**	**Recall**	F_1
Text Bi-LSTM	Argument	0.816 ± 0.027	0.946 ± 0.022	0.876 ± 0.015	0.686 ± 0.045	0.749 ± 0.025	0.715 ± 0.031
	Other	0.669 ± 0.067	0.332 ± 0.092	0.436 ± 0.083	0.723 ± 0.027	0.656 ± 0.061	0.687 ± 0.044
	Wt. average	0.781 ± 0.029	0.797 ± 0.024	0.769 ± 0.034	0.705 ± 0.033	0.702 ± 0.035	0.701 ± 0.036
	Macro av.	0.743 ± 0.037	0.639 ± 0.039	0.656 ± 0.046	0.704 ± 0.033	0.702 ± 0.035	0.701 ± 0.036
Text BERT	Argument	0.784 ± 0.042	0.985 ± 0.016	0.873 ± 0.020	0.571 ± 0.117	0.712 ± 0.231	0.610 ± 0.101
	Other	0.391 ± 0.395	0.148 ± 0.192	0.206 ± 0.262	0.535 ± 0.204	0.441 ± 0.352	0.438 ± 0.317
	Wt. average	0.687 ± 0.133	0.782 ± 0.040	0.710 ± 0.082	0.552 ± 0.155	0.570 ± 0.117	0.520 ± 0.174
	Macro av.	0.588 ± 0.218	0.567 ± 0.089	0.539 ± 0.140	0.553 ± 0.152	0.577 ± 0.109	0.524 ± 0.168
Audio	Argument	0.806 ± 0.045	0.782 ± 0.419	0.708 ± 0.360	0.614 ± 0.043	0.658 ± 0.256	0.607 ± 0.131
	Other	0.477 ± 0.380	0.310 ± 0.400	0.237 ± 0.182	0.670 ± 0.112	0.549 ± 0.238	0.563 ± 0.130
	Wt. average	0.731 ± 0.079	0.671 ± 0.232	0.597 ± 0.258	0.639 ± 0.048	0.613 ± 0.023	0.590 ± 0.036
	Macro av.	0.642 ± 0.183	0.546 ± 0.050	0.472 ± 0.160	0.642 ± 0.051	0.604 ± 0.027	0.585 ± 0.041
Multimodal (Bi-LSTM +Audio)	Argument	0.684 ± 0.382	0.635 ± 0.360	0.657 ± 0.369	0.755 ± 0.140	0.386 ± 0.260	0.445 ± 0.258
	Other	0.412 ± 0.103	0.641 ± 0.213	0.472 ± 0.054	0.612 ± 0.059	0.834 ± 0.131	0.697 ± 0.027
	Wt. average	0.620 ± 0.316	0.637 ± 0.227	0.614 ± 0.294	0.679 ± 0.044	0.625 ± 0.052	0.581 ± 0.113
	Macro av.	0.548 ± 0.241	0.638 ± 0.078	0.565 ± 0.210	0.684 ± 0.052	0.610 ± 0.066	0.571 ± 0.124
Multimodal (BERT +Audio)	Argument	0.833 ± 0.030	0.936 ± 0.023	0.881 ± 0.009	0.722 ± 0.017	0.789 ± 0.033	0.753 ± 0.012
	Other	0.710 ± 0.060	0.445 ± 0.095	0.538 ± 0.056	0.771 ± 0.021	0.699 ± 0.037	0.732 ± 0.017
	Wt. average	0.803 ± 0.012	0.810 ± 0.014	0.793 ± 0.024	0.747 ± 0.009	0.743 ± 0.009	0.743 ± 0.010
	Macro av.	**0.771 ± 0.017**	**0.690 ± 0.034**	**0.709 ± 0.030**	**0.746 ± 0.009**	**0.744 ± 0.009**	**0.743 ± 0.010**

Table 2: Models' performance for 10% of the datasets. Errors indicate standard deviation after 5 replicates.

averaged F-score, with 0.624. Our models perform slightly better than those of Mancini et al. (2022), who used the same architecture (although with their own hyperparameter optimization) and dataset (although slightly smaller). But, like us, they find that the multimodal models do not significantly (if at all) outperform the text-only models, with macro F-scores of 0.674 in both. Their audio-only model was not better than the random baseline at 0.505.

In both the original work of Mestre et al. (2021) and the replication by Mancini et al. (2022), the authors show a beneficial impact of audio embeddings in argument classification. However, that dataset was significantly smaller ($N = 4,104$) and heavily imbalanced towards one of the classes. Our dataset is also slightly imbalanced towards the "argument" class (76.15% arguments) and the text-only models seem to be reaching saturation, as per the previous paragraph. Moreover, the low precision and recall of the "other" class leads us to believe that the models overly rely on classifying many instances as "argument". Therefore, we asked ourselves what would happen: 1) when the dataset is small and the performance of the text-only model might suffer; 2) when the dataset is balanced and the models cannot rely on over-classifying the "argument" class. Does the addition of audio improve the performance metrics in those cases?

4.2 Fractional and balanced datasets

The right-hand side of Table 1 shows the results with the same models for a balanced dataset. To obtain a balanced dataset, a random number of "argument" classes were dropped from the original dataset, until we obtained an equal number of both classes, therefore reducing the total size to 13,758 sentences. This table shows how the overall performance of the BERT and Bi-LSTM models has been reduced, reaching macro averaged F-score values of 0.754 for the BERT model and 0.728 for the Bi-LSTM model. Moreover, the precision and recall of both classes are more balanced: whereas in the original dataset the recall of the "argument" and "other" classes of the BERT model were 0.951 and 0.487, respectively, they are now 0.839 and 0.674. The multimodal model continues to perform significantly better than the audio-only model, but still not better than the BERT model, to which it still achieves similar F-score values. It seems, therefore, that a multimodal model does not provide better results than text-only models when the datasets are balanced, at least as long as the number of annotations continues to be large ($N = 13,758$). The text-only models still seem to reach saturation of what can be accomplished with the dataset.

Table 2 shows the results for a fractional dataset composed of only 10% of the original and balanced data. We hypothesize that the performance of the text-only models will start to suffer with small amounts of training data and the audio features from the multimodal models will be able to partially recover previous performance. Indeed, it has been shown in some work that BERT models tend to decline in performance with small datasets and can be outperformed by simpler models like Bi-LSTM (Ezen-Can, 2020). Likewise, not only does the overall performance suffer, but also the stability of the model as discussed by Dodge et al. (2020). In our case, we see that the BERT model shows a large drop in macro average F-score, down to 0.539, and high instability, as can be observed from the large standard error of the "other" class. In this case, the Bi-LSTM model outperforms the BERT model at 0.656, suggesting that the BERT model is more sensitive to smaller datasets. The BERT-based multimodal model also outperforms the BERT model, with a macro average F-score of 0.709, very close to the best scenario with the original dataset at 0.746. Surprisingly, the Bi-LSTM-based multimodal model does not outperform the Bi-LSTM model, but worsens its performance. On 10% of the balanced dataset, results are similiar, with the BERT model suffering and the BERT-based multimodal model outperforming all with F-scores of 0.743, close to the original balanced case.

For intermediate sizes, such as 50% and 20% (reported in Section C) the change seems to be gradual. For both 50% and 20% of the original dataset, the performance of the only-text BERT model and the BERT-based multimodal model is practically identical. However, for balanced datasets of 20% (with only $N = 2,751$), the performance of the BERT model starts to decrease and is overcome by the multimodal model. All these results suggest that there is a point at around $N = 3,000$ or below where audio features provide an important added value in the performance of the models. Full details of all the replicates, including the model history with validation losses and accuracy, confusion matrices and a full breakdown of the performance metrics can be found in the repository of the project[5] and its official release (Mestre et al., 2023).

5 Artificial voices

It is not clear what the audio models are specifically looking at when they undergo training, as the features extracted are not always easy to interpret. One hypothesis is that they learn from the words being uttered, their pronunciation and associations thereof in a similar way to text-based models. However, it is also likely that these audio models are picking up intonation or pitch features that are possibly different when one person is making an argument.

As a first step in investigating what the audio-based models are paying attention to, we ran our models using artificial voices, instead of the original voices from the presidential candidates. For this, we used the computer generated voices from the Microsoft Window's Text to Speech (TTS) system. Then, we used the text-to-speech conversion library pyttsx3 (v 2.9) for Python to run each sentence through the Microsoft TTS system and generate utterances spoken by the so-called Microsoft Mark and Microsoft Zira, the male and female version of US voices. We set the speaking rate at 200 words per minute, but it could be interesting for future studies to observe the accuracy of the models when the speech rate is changed. Likewise, each country package in Microsoft Windows has its own set of unique voices, even if they speak the same language, e.g., UK, India, South Africa, and so on, so it could be interesting to check potential differences in accuracy with different accents.

We then ran our audio-only models as described before and the performance metrics are displayed in Table 3. We only report the F-scores, and not precision and recall, for simplicity.[6] We can see in this table that, generally, there are no differences in F-score by gender of the artificial voices, although there seems to be a bias towards the female Zira voice. When compared to the audio model run with original voices, it is interesting to see that whereas using artificial voices results in a similar score with the full dataset, with the balanced dataset the results are improved, going from 0.566 as reported in 1 to 0.626 using the Zira voice. This improvement is also found with the 10% dataset, which reported 0.536 and 0.594 for the original and balanced datasets, whereas Table 2 reported 0.572 and 0.585, respectively. In the balanced case, these values are even better than the BERT model at 0.524. A potential explanation is that the artificial audio models lack noise coming from the recording, or remove variation coming from different people having different baseline pitches that might confuse the model. There might be a trade-off between taking advantage of pitch or intonation during arguments (which we were not able to prove produces any effect) and benefiting from the noise-reduced nature of artificial audio. In any case, it seems that audio-only models based on artificial voices can learn features and classify arguments with an accuracy comparable to text-only models, and sometimes

[5] `https://github.com/rafamestre/Multimodal-USElecDeb60To16`.

[6] Full information, all the computerized utterances and scripts to reproduce our results can be found in our repository (Mestre et al., 2023).

Voice	Class	F_1			
		Original dataset	**Balanced dataset**	**10% original**	**10% balanced**
Female	Argument	0.874 ± 0.002	0.596 ± 0.141	0.865 ± 0.012	0.555 ± 0.055
	Other	0.235 ± 0.029	0.656 ± 0.017	0.207 ± 0.115	0.633 ± 0.051
	Wt. average	0.722 ± 0.006	0.626 ± 0.064	0.706 ± 0.024	0.593 ± 0.025
	Macro av.	$\mathbf{0.555 \pm 0.014}$	$\mathbf{0.626 \pm 0.063}$	$\mathbf{0.536 \pm 0.054}$	$\mathbf{0.594 \pm 0.022}$
Male	Argument	0.871 ± 0.003	0.582 ± 0.126	0.875 ± 0.007	0.496 ± 0.173
	Other	0.199 ± 0.043	0.654 ± 0.167	0.191 ± 0.121	0.595 ± 0.127
	Wt. average	0.711 ± 0.010	0.618 ± 0.060	0.720 ± 0.032	0.548 ± 0.057
	Macro av.	0.535 ± 0.021	0.618 ± 0.059	0.533 ± 0.062	0.545 ± 0.059

Table 3: Audio-only models' performance with artificial voices. Errors indicate standard deviation after 5 replicates.

even better than those when data is scarce. Models based on the original voices often struggle, and this might be due to the inherent noise of the recordings or differences at the speaker level.

6 Ablation study

Finally, to further understand how the different LLD audio features might play a role in argument detection, we perform an ablation study with the audio model in which we eliminate one of the six features at a time and assess how the performance of the model changes. The results are in Table 4 for the full and balanced datasets (we only report F-score for simplicity). The first column indicates the feature that was eliminated from the feature tensor, which originally had dimensions of $(45, T_{max})$, whereas the second column displays the dimensions of the tensor after elimination. In general, none of the cases deviate much from the full-feature model with macro F-scores of 0.540 and 0.566 for the original and balanced datasets (Table 1). The first four features are spectral features, meaning that they are features extracted from the spectrogram of the sound wave. In particular, the spectral centroid and bandwidth characterize the center of mass of the spectrum (where most of the energy is located) and its weighted standard deviation, respectively (Sandhya et al., 2020). The spectral rolloff also characterizes the energy spectrum by identifying the frequency below which a certain percentage (in our case, 85%) of the energy is located, and can be used to differentiate voices from noise (Syed et al., 2021). These three features are 1-dimensional and only reduce the feature space to $(44, T)$. The spectral contrast feature, however, is 7-dimensional and works by dividing the spectrogram into 6 sub-bands, for which the difference between their peaks and valleys are computed, and is commonly used in music identification (Jiang et al., 2002b). The chromagram, or chroma feature, is a feature that aggregates all information of a waveform into the 12 different pitch classes, which are separated by an octave. This feature is mostly used for music synchronization and singing voice separation (Yuan et al., 2022). Finally, MFCCs are a variable set of features (in our case, we use 12) which describe the shape of the spectral signal. They are based on human auditive perceptions and are widely used in the literature to capture phonetically relevant features (Mansour and Lachiri, 2017).

From the ablation study, we see that skipping features does not have a strong influence on the performance with the original full dataset. With the balanced dataset, the elimination of the spectral roll-off feature seems to have a strong effect, as it decreases its macro F-score to 0.402. A special mention to the MFCCs is deserved. These are the most common LDDs in the literature. Eliminating this feature but keeping the rest does not affect the performance (F-score of 0.544) in the original dataset, and improves it in the balanced case to 0.612. As MFCCs (and many of the other features) are commonly used to distinguish between voices based on their frequency and pitch, they could bias the model by considering information about the speaker, which is not necessarily relevant to the argument. This would also explain why the artificial voices performed better than the original dataset and why some of the simplest features, like spectral roll-off, have a big influence on performance.

7 Conclusion

In this paper, we explored the possibilities of using audio to detect arguments with multimodal machine learning models in a dataset of US presidential debates that was annotated for arguments. We found that, generally, BERT-based text-only models outperformed all models in the original dataset, but multimodal models can improve performance

Feature skipped	Feature space	Class	F_1	
			Original dataset	**Balanced dataset**
Spectral centroids	(44,T)	Argument	0.858 ± 0.010	0.511 ± 0.185
		Other	0.225 ± 0.156	0.613 ± 0.005
		Wt. average	0.706 ± 0.035	0.561 ± 0.070
		Macro av.	0.542 ± 0.075	0.562 ± 0.069
Spectral bandwidth	(44,T)	Argument	0.869 ± 0.001	0.360 ± 0.270
		Other	0.198 ± 0.018	0.653 ± 0.039
		Wt. average	0.709 ± 0.001	0.508 ± 0.122
		Macro av.	0.534 ± 0.009	0.507 ± 0.122
Spectral roll-off	(44,T)	Argument	0.777 ± 0.109	0.127 ± 0.118
		Other	0.320 ± 0.192	0.677 ± 0.007
		Wt. average	0.668 ± 0.065	0.408 ± 0.058
		Macro av.	**0.548 ± 0.075**	0.402 ± 0.059
Spectral contrast	(38,T)	Argument	0.866 ± 0.002	0.429 ± 0.278
		Other	0.201 ± 0.110	0.645 ± 0.025
		Wt. average	0.706 ± 0.028	0.537 ± 0.127
		Macro av.	0.533 ± 0.055	0.537 ± 0.128
Chroma	(33,T)	Argument	0.870 ± 0.003	0.408 ± 0.316
		Other	0.196 ± 0.038	0.611 ± 0.061
		Wt. average	0.710 ± 0.012	0.509 ± 0.133
		Macro av.	0.533 ± 0.020	0.509 ± 0.132
MFCCs	(22,T)	Argument	0.869 ± 0.003	0.619 ± 0.032
		Other	0.220 ± 0.018	0.605 ± 0.058
		Wt. average	0.714 ± 0.011	0.611 ± 0.022
		Macro av.	**0.544 ± 0.011**	**0.612 ± 0.021**

Table 4: Results from ablation study. The complete feature space has a dimensions of $(45, T_{max})$, where T_{max} is the 99% percentile length of the utterances.

when the datasets are small and BERT encodings start performing poorly, in both balanced and unbalanced versions of the data. Multimodal models are therefore an alternative that could be used to improve classifications of arguments when data is scarce. To further investigate the reasons for these improvements, we ran audio-only models using artificially generated voices of male and female genders. Although we did not find a significant difference in the performance with the artificial voices (only a slight preference toward female), we find that in the most difficult scenarios (balanced and small datasets), the models with artificial voices outperform those with the original audio from the debates. Moreover, we perform an ablation study and we find that removing certain features like MFCCs can improve the performance of the models. We recognize that these features are commonly used to distinguish between speakers, so irrelevant characteristics of speakers might be influencing the capacity of the models to accurately classify arguments. However, all these features are highly correlated with one another, so further work should investigate which features are more independent of the speakers themselves or if they can be normalized before being fed into the network. These results should be compared with artificially generated voices, which could have different accents or speech rates to understand how those features can influence the classification of arguments.

Limitations

This paper assesses the benefits of audio features in the task of argumentation mining. Although the dataset presented has a large number of annotations ($N = 28,850$) further research should be aimed at studying its cross-domain adaptability in different scenarios and datasets. The model architectures used in this paper are fairly standard in the literature and thus represent benchmarking results for further research in the field, as the application of multimodal sources of data in argumentation mining is still largely unexplored. Newer architectures, for instance those based on cross-attention mechanisms between the different modalities, should be explored next to check whether these results could be improved. It is still not fully clear what information from the audio embeddings are being picked up by the models. Our study on computerized voices offers an interesting avenue of research, but it should be further expanded to include a larger array of voices, different speech rates and embedding in fully multimodal models to assess their performance. The potential biases of models that use audio, especially how different voice's characteristics (such as pitch frequency, which is correlated to gender) affect the classification, are not fully studied here, but briefly touched upon in the computerized voice study. This is rather important but unexplored territory and normalization strategies should be investigated to solve these issues. Finally, there are a large number of audio features that could be explored in this domain. Those used in this work are just some of the most common ones, but, as mentioned before, the choice is generally based on the extraction package used by the researcher. There is a need for standardization of these features in the community, so different work can be better compared.

Ethics Statement

We acknowledge that the use of audio features for automatic classification can cause potential privacy issues, as the real voice, and not only the speech, is used for classification. At this stage, we do not foresee any outstanding ethical issues from this research, as we use public domain data that was televised in public national television and is widely availabe on the web. Ethics approval for this research was received from the University of Southampton's Faculty of Social Science Ethics and Research Governance committee, Ref: 66226, Date 16/03/2022. The audio-enhanced dataset of this work along with the codes to reproduce our results are made available under license CC0: Public Domain in the project's GitHub and is also assigned a DOI to ensure reliable access to this work's supplementary data (Mestre et al., 2023).

Acknowledgements

This work has been funded by UK Research and Innovation (UKRI) funding (grant ref MR/S032711/1) and by the Web Science Institute of the University of Southampton (project PP-2020-Mestre). This work was also supported by the Natural Environment Research Council (NE/S015604/1) and Economic and Social Research Council (ES/V011278/1). The authors acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work.

References

Teresa Alsinet, Josep Argelich, Ramón Béjar, and Joel Cemeli. 2019. A distributed argumentation algorithm for mining consistent opinions in weighted twitter discussions. *Soft Computing*, 23:2147–2166.

Bagus Tris Atmaja and Masato Akagi. 2020. Dimensional speech emotion recognition from speech features and word embeddings by using multitask learning. *APSIPA Transactions on Signal and Information Processing*, 9.

Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2018. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24.

James Bergstra, Daniel Yamins, and David Cox. 2013. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*, pages 115–123. PMLR.

C. Busso, M. Bulut, C. C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan. 2008. IEMOCAP: Interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42(4):335–359.

Linqin Cai, Yaxin Hu, Jiangong Dong, and Sitong Zhou. 2019. Audio-textual emotion recognition based on improved neural networks. *Mathematical Problems in Engineering*, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

Florian Eyben, Klaus R. Scherer, Bjorn W. Schuller, Johan Sundberg, Elisabeth Andre, Carlos Busso, Laurence Y. Devillers, Julien Epps, Petri Laukka, Shrikanth S. Narayanan, and Khiet P. Truong. 2016. The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing. *IEEE Transactions on Affective Computing*, 7:190–202.

Florian Eyben, Felix Weninger, Florian Gross, and Björn Schuller. 2013. Recent developments in opensmile, the munich open-source multimedia feature extractor. pages 835–838.

Aysu Ezen-Can. 2020. A comparison of lstm and bert for small corpus. *arXiv preprint arXiv:2009.05451*.

Debanjan Ghosh, Smaranda Muresan, Nina Wacholder, Mark Aakhus, and Matthew Mitsui. 2014. Analyzing argumentative discourse units in online interactions. In *Proceedings of the first workshop on argumentation mining*, pages 39–48.

Theodoros Giannakopoulos. 2015. Pyaudioanalysis: An open-source python library for audio signal analysis. *PLoS ONE*, 10.

Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. 2010. Multimodal semi-supervised learning for image classification. pages 902–909.

Shohreh Haddadan, Elena Cabrio, and Serena Villata. 2019. Yes, we can! mining arguments in 50 years of us presidential campaign debates. In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*, pages 4684–4690.

Ngoc Huynh Ho, Hyung Jeong Yang, Soo Hyung Kim, and Gueesang Lee. 2020. Multimodal approach of speech emotion recognition using multi-level multi-head fusion attention-based recurrent neural network. *IEEE Access*, 8:61672–61686.

Anurag Illendula and Amit Sheth. 2019. Multimodal emotion classification.

Philip Jackson and SJUoSG Haq. 2014. Surrey audio-visual expressed emotion (savee) database. *University of Surrey: Guildford, UK*.

Rajiv Jain and Curtis Wigington. 2019. Multimodal document image classification. pages 71–77. IEEE Computer Society.

D. N. Jiang, L. Lu, H. J. Zhang, J. H. Tao, and L. H. Cai. 2002a. Music type classification by spectral contrast feature. *Proc. IEEE Int. Conf. on Multimedia and Expo*, pages 113–116.

Dan-Ning Jiang, Lie Lu, Hong-Jiang Zhang, Jian-Hua Tao, and Lian-Hong Cai. 2002b. Music type classification by spectral contrast feature. In *Proceedings. IEEE International Conference on Multimedia and Expo*, volume 1, pages 113–116. IEEE.

A. Klapuri and M. Davy. 2006. Signal processing methods for music transcription. In *Signal Processing Methods for Music Transcription*, chapter 5. Springer Science & Business Media.

John Lawrence and Chris Reed. 2017. Using complex argumentative interactions to reconstruct the argumentative structure of large-scale debates. In *Proceedings of the 4th Workshop on Argument Mining*, pages 108–117.

Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Moritz Hardt, Ben Recht, and Ameet Talwalkar. 2018. Massively parallel hyperparameter tuning.

Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E Gonzalez, and Ion Stoica. 2018. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*.

Marco Lippi and Paolo Torroni. 2016. Argument mining from speech: Detecting claims in political debates. *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pages 2979–2985.

Steven R Livingstone and Frank A Russo. 2018. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5):e0196391.

Eleonora Mancini, Federico Ruggeri, Andrea Galassi, and Paolo Torroni. 2022. Multimodal argument mining: A case study in political debates. pages 158–170.

Asma Mansour and Zied Lachiri. 2017. Svm based emotional speaker recognition using mfcc-sdc features. *International Journal of Advanced Computer Science and Applications*, 8(4).

Brian McFee, Colin Raffel, Dawen Liang, Daniel P Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, pages 18–25.

Rafael Mestre, Stuart E. Middleton, Matt Ryan, Masood Gheasi, Timothy J. Norman, and Jiatong Zhu. 2023. rafamestre/multimodal-uselecdeb60to16: v1.0.0.

Rafael Mestre, Razvan Milicin, Stuart Middleton, Matt Ryan, Jiatong Zhu, and Timothy J Norman. 2021. M-arg: Multimodal argument mining dataset for political debates with audio and transcripts. In *Proceedings of the 8th Workshop on Argument Mining*, pages 78–88.

Louis-Philippe Morency and Tadas Baltrušaitis. 2017. Multimodal machine learning: integrating language, vision and speech. In *Proceedings of the 55th annual meeting of the association for computational linguistics: Tutorial abstracts*, pages 3–5.

Kai Nakamura, Sharon Levy, and William Yang Wang. 2020. r/fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection. pages 11–16.

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

P Sandhya, V Spoorthy, Shashidhar G Koolagudi, and NV Sobhana. 2020. Spectral features for emotional speaker recognition. In *2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAECC)*, pages 1–6. IEEE.

Prabhav Singh, Ridam Srivastava, K. P.S. Rana, and Vineet Kumar. 2021. A multimodal hierarchical approach to speech emotion recognition from audio and text[formula presented]. *Knowledge-Based Systems*, 229.

C. Stab and I. Gurevych. 2014. Annotating argument components and relations in persuasive essays. In *Proc. 25th Int. Conf. on Computational Linguistics*, pages 1501–1510.

S Syed, Munaf Rashid, Samreen Hussain, Anoshia Imtiaz, Hamnah Abid, and Hira Zahid. 2021. Inter classifier comparison to detect voice pathologies. *Mathematical Biosciences and Engineering*, 18(3):2258–2273.

Jacky Visser, John Lawrence, Chris Reed, Jean Wagemans, and Douglas Walton. 2021. *Annotating Argument Schemes*, volume 35.

Yue Xie, Ruiyu Liang, Zhenlin Liang, Chengwei Huang, Cairong Zou, and Bjorn Schuller. 2019. Speech emotion classification using attention-based lstm. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 27:1675–1685.

Bo Yang, Bo Shao, Lijun Wu, and Xiaola Lin. 2022. Multimodal sentiment analysis with unidirectional modality translation. *Neurocomputing*, 467:130–137.

Zengwei Yao, Zihao Wang, Weihuang Liu, Yaqian Liu, and Jiahui Pan. 2020. Speech emotion recognition using fusion of three multi-task learning-based classifiers: Hsf-dnn, ms-cnn and lld-rnn. *Speech Communication*, 120:11–19.

Seunghyun Yoon, Seokhyun Byun, and Kyomin Jung. 2018. Multimodal speech emotion recognition using audio and text. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 112–118. IEEE.

Siyuan Yuan, Zhepei Wang, Umut Isik, Ritwik Giri, Jean-Marc Valin, Michael M Goodwin, and Arvindh Krishnaswamy. 2022. Improved singing voice separation with chromagram-based pitch-aware remixing. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 111–115. IEEE.

Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. 2016. Multimodal sentiment intensity analysis in videos: Facial gestures and verbal messages. *IEEE Intelligent Systems*, 31(6):82–88.

Jianfeng Zhao, Xia Mao, and Lijiang Chen. 2019. Speech emotion recognition using deep 1d 2d cnn lstm networks. *Biomedical Signal Processing and Control*, 47:312–323.

A Hyperparameter tuning

Table A1 shows the hyperparameter bounds, as well as the final values selected for each model. After hyperparameter tuning, BERT-based text-models took approximately 45 minutes to train, whereas Bi-LSTM-based text-models took only 2 minutes (on the full dataset). Audio-only models took 25 min to train (as they would converge very slowly taking more than 50 training epochs), BERT-based multimodal models took an average of 3 hours and Bi-LSTM-based multimodal models took 8 minutes.

B Notes on alignment

As mentioned in the main text, we used the Aeneas package, which is a Python package that uses “forced alignment” to match the text and audio from utterances. Briefly, each sentence is provided as text and the tool uses the *espeak* Windows speech synthesizer to generate a computerized voice uttering that sentence. Then, amplitude waves are contrasted from the real and computer-generated sentences, and they are aligned to extract the timestamps. Two researchers manually checked every debate for major misalignment (and fixed them) until we obtained an almost perfectly aligned text.

We indicated omissions to be ignored during alignment: text between brackets that indicates transcription tags like "applause", or the interjection "uh", that appeared (too much) in some of the transcripts and never in others. This tool comes with a handy HTML output that allows the user to click on different parts of the transcript and check the alignment. Two people manually checked every debate for major misalignment (and fixed those cases as described above) until we obtained an almost perfectly aligned text.

Together with the dataset and codes to reproduce the results, we present our code to reproduce the alignment process, as well as an exhaustive list of the problems we encountered during alignment and how we solved them (e.g., modifications to the original transcripts, splitting videos, etc.). We share a folder with all the results from training our models with original and balanced datasets, as well as fractional subsets of 50%, 20% and 10%. We also include the training with artificial voices and from the ablation study. Each subfolder contains the parameters used by the model, confusion matrices of each run (5 runs per model), loss value vs epoch plots, training history with validation metrics, and precision/recall/F-score metrics for each run, as well as the average values.

C Results for 50% and 20% datasets

Tables A2 and A3 show the results for all models with 50% and 20% fractional datasets.

Hyper-parameter	**Range**	**Bi-LSTM**	**BERT**	**Audio**	**Multimodal (Bi-LSTM +Audio)**	**Multimodal (BERT +Audio)**
Learning rate	[0.01, 0.000001]	6.68e−4	2.37e−6	1.89e−5	7.73e−5	2.81e−6
Batch size	{16, 32, 64}	16	32	64	16	16
Hidden activation	{ReLU,Sigm., Tanh}	Tanh	ReLU	Sigm.	ReLU	ReLU
Trainable	{True, False}	False	True		True	True
Dropout text	[0, 0.9]	0.6	0.8		0.5	0.7
Dropout audio	[0, 0.9]			0.6	0.6	0.1
Dropout final	[0, 0.9]				0	0.5
# neurons dense layer	{16, 32, 64, 128, 256}	32	256	32	16	16
# neurons Bi-LSTM text	{16, 32, 64, 128, 256}	64			64	
# neurons Bi-LSTM audio	{16, 32, 64, 128, 256}			16	32	256
# filters conv. layer 1	{4, 8, 16, 32, 64}			4	4	8
# filters conv. layer 2	{4, 8, 16, 32, 64}			4	8	4
Kernel size conv. layer 1	{1, 3, 5, 7}			3	3	3
Kernel size conv. layer 2	{1, 3, 5, 7}			5	7	3
Size pooling layer 1	{2, 4}			2	2	2
Size pooling layer 2	{2, 4}			2	2	4
Number of parameters		2,802,074	109,679,619	5,362,110	7,928,618	119,978,775

Table A1: List of hyperparameters, their search range and their optimal value for each model. The range in "learning rate" (in squared brackets) was given as a log uniform distribution, in the dropout layers a uniform distribution in multiples of 0.1, whereas in the remaining cases (represented with curly brackets) the choices were from the discrete set of values shown.

Model	**Class**	50% original dataset ($N = 14,425$)			50% balanced dataset ($N = 6,879$)		
		Precision	**Recall**	F_1	**Precision**	**Recall**	F_1
Text Bi-LSTM	Argument	0.836 ± 0.014	0.944 ± 0.012	0.886 ± 0.005	0.704 ± 0.025	0.784 ± 0.027	0.741 ± 0.004
	Other	0.705 ± 0.044	0.415 ± 0.033	0.521 ± 0.019	0.760 ± 0.020	0.673 ± 0.049	0.713 ± 0.026
	Wt. average	0.805 ± 0.006	0.816 ± 0.007	0.798 ± 0.011	0.732 ± 0.008	0.728 ± 0.013	0.727 ± 0.014
	Macro av.	0.770 ± 0.016	0.680 ± 0.011	0.704 ± 0.011	0.732 ± 0.008	0.728 ± 0.012	0.727 ± 0.014
Text BERT	Argument	0.841 ± 0.041	0.952 ± 0.028	0.892 ± 0.013	0.719 ± 0.010	0.803 ± 0.023	0.759 ± 0.009
	Other	0.776 ± 0.127	0.402 ± 0.224	0.474 ± 0.262	0.773 ± 0.018	0.680 ± 0.027	0.723 ± 0.016
	Wt. average	0.826 ± 0.004	0.822 ± 0.031	0.794 ± 0.071	0.746 ± 0.010	0.742 ± 0.010	0.741 ± 0.010
	Macro av.	$\mathbf{0.809 \pm 0.043}$	0.677 ± 0.098	0.683 ± 0.137	$\mathbf{0.746 \pm 0.010}$	$\mathbf{0.742 \pm 0.010}$	$\mathbf{0.741 \pm 0.010}$
Audio	Argument	0.795 ± 0.032	0.897 ± 0.176	0.833 ± 0.079	0.630 ± 0.030	0.521 ± 0.211	0.547 ± 0.122
	Other	0.398 ± 0.246	0.211 ± 0.253	0.216 ± 0.153	0.601 ± 0.067	0.681 ± 0.164	0.623 ± 0.050
	Wt. average	0.703 ± 0.066	0.738 ± 0.078	0.690 ± 0.042	0.616 ± 0.022	0.599 ± 0.034	0.584 ± 0.050
	Macro av.	0.597 ± 0.126	0.554 ± 0.042	0.524 ± 0.052	0.615 ± 0.023	0.601 ± 0.029	0.585 ± 0.048
Multimodal (Bi-LSTM +Audio)	Argument	0.870 ± 0.036	0.756 ± 0.190	0.794 ± 0.113	0.800 ± 0.096	0.330 ± 0.282	0.398 ± 0.271
	Other	0.493 ± 0.119	0.622 ± 0.181	0.521 ± 0.049	0.582 ± 0.091	0.887 ± 0.117	0.692 ± 0.029
	Wt. average	0.780 ± 0.013	0.722 ± 0.106	0.728 ± 0.092	0.694 ± 0.027	0.603 ± 0.091	0.542 ± 0.154
	Macro av.	0.682 ± 0.044	0.689 ± 0.034	0.657 ± 0.073	0.691 ± 0.023	0.609 ± 0.084	0.545 ± 0.149
Multimodal (BERT +Audio)	Argument	0.842 ± 0.010	0.946 ± 0.011	0.891 ± 0.223	0.727 ± 0.025	0.778 ± 0.028	0.751 ± 0.020
	Other	0.736 ± 0.037	0.457 ± 0.029	0.563 ± 0.014	0.756 ± 0.021	0.701 ± 0.022	0.727 ± 0.010
	Wt. average	0.816 ± 0.002	0.825 ± 0.003	0.810 ± 0.005	0.742 ± 0.013	0.740 ± 0.014	0.739 ± 0.013
	Macro av.	0.789 ± 0.014	$\mathbf{0.701 \pm 0.009}$	$\mathbf{0.727 \pm 0.006}$	0.741 ± 0.014	0.739 ± 0.013	0.739 ± 0.013

Table A2: Models' performance for 50% of the datasets. Errors indicate standard deviation after 5 replicates.

Model	**Class**	20% original dataset ($N = 5,770$)			20% balanced dataset ($N = 2,751$)		
		Precision	**Recall**	F_1	**Precision**	**Recall**	F_1
Text Bi-LSTM	Argument	0.833 ± 0.012	0.941 ± 0.026	0.883 ± 0.012	0.699 ± 0.024	0.723 ± 0.064	0.709 ± 0.032
	Other	0.679 ± 0.066	0.386 ± 0.043	0.488 ± 0.019	0.728 ± 0.038	0.701 ± 0.053	0.713 ± 0.028
	Wt. average	0.797 ± 0.015	0.810 ± 0.015	0.790 ± 0.013	0.714 ± 0.023	0.712 ± 0.024	0.711 ± 0.024
	Macro av.	0.756 ± 0.030	0.663 ± 0.010	0.686 ± 0.009	0.714 ± 0.023	0.712 ± 0.040	0.711 ± 0.024
Text BERT	Argument	0.844 ± 0.015	0.952 ± 0.015	0.895 ± 0.023	0.623 ± 0.112	0.691 ± 0.166	0.648 ± 0.128
	Other	0.757 ± 0.047	0.453 ± 0.026	0.566 ± 0.021	0.648 ± 0.117	0.570 ± 0.184	0.598 ± 0.147
	Wt. average	0.823 ± 0.019	0.831 ± 0.017	0.815 ± 0.018	0.635 ± 0.113	0.632 ± 0.116	0.624 ± 0.122
	Macro av.	$\mathbf{0.801 \pm 0.026}$	$\mathbf{0.703 \pm 0.013}$	$\mathbf{0.731 \pm 0.016}$	0.635 ± 0.113	0.631 ± 0.115	0.623 ± 0.122
Audio	Argument	0.802 ± 0.024	0.770 ± 0.246	0.766 ± 0.131	0.410 ± 0.422	0.207 ± 0.422	0.155 ± 0.279
	Other	0.453 ± 0.157	0.365 ± 0.289	0.304 ± 0.132	0.575 ± 0.127	0.817 ± 0.384	0.589 ± 0.205
	Wt. average	0.718 ± 0.026	0.677 ± 0.121	0.657 ± 0.079	0.488 ± 0.237	0.519 ± 0.031	0.378 ± 0.051
	Macro av.	0.628 ± 0.068	0.568 ± 0.028	0.535 ± 0.037	0.492 ± 0.240	0.512 ± 0.020	0.372 ± 0.048
Multimodal (Bi-LSTM +Audio)	Argument	0.873 ± 0.089	0.432 ± 0.398	0.423 ± 0.429	0.751 ± 0.045	0.457 ± 0.118	0.558 ± 0.099
	Other	0.357 ± 0.107	0.792 ± 0.200	0.467 ± 0.070	0.610 ± 0.039	0.841 ± 0.068	0.705 ± 0.026
	Wt. average	0.748 ± 0.073	0.522 ± 0.254	0.473 ± 0.342	0.681 ± 0.022	0.649 ± 0.037	0.631 ± 0.055
	Macro av.	0.615 ± 0.067	0.612 ± 0.102	0.470 ± 0.248	0.680 ± 0.022	0.649 ± 0.035	0.631 ± 0.054
Multimodal (BERT +Audio)	Argument	0.843 ± 0.005	0.955 ± 0.011	0.896 ± 0.006	0.714 ± 0.014	0.793 ± 0.047	0.751 ± 0.019
	Other	0.745 ± 0.055	0.422 ± 0.027	0.538 ± 0.031	0.761 ± 0.045	0.673 ± 0.288	0.713 ± 0.019
	Wt. average	0.820 ± 0.014	0.830 ± 0.010	0.812 ± 0.010	0.738 ± 0.021	0.734 ± 0.017	0.732 ± 0.017
	Macro av.	0.794 ± 0.028	0.689 ± 0.015	0.717 ± 0.017	$\mathbf{0.738 \pm 0.021}$	$\mathbf{0.733 \pm 0.018}$	$\mathbf{0.732 \pm 0.017}$

Table A3: Models' performance for 20% of the datasets. Errors indicate standard deviation after 5 replicates.