Sentiment Analysis of Incoming Calls on Helpdesk

1st Deepika C S
Department of CSE & IS,
Presidency University
Bengaluru, India
Deepika.20211COM0075@presidencyuniversity.in

2nd Impa B H
Department of CSE & IS
Presidency University
Bengaluru, India
impa@presidencyuniversity.in

3rd Prakash Singh
Department of CSE & IS
Presidency University
Bengaluru, India
Prakash.20211CEI0055@presidencyuniversity.in

4th Dishik L Setty
Department of CSE & IS
Presidency University
Bengaluru, India
Dishik.20211COM0006@presidencyuniversit y.in

5th Paavana Gowda
Department of CSE & IS
Presidency University
Bengaluru, India
Paavana.20211COM0029@presidencyuniversity.in

Abstract *- **This work introduces a holistic framework for audio data processing and analysis via an integrated web service. The system provides a multi-purpose pipeline optimized for noisy and multilingual inputs by utilizing state-of-the-art models like FasterWhisper for transcription, DeepFilterNet for denoising audio, and transformer-based models for sentiment and sarcasm detection. The backend uses Flask to provide modular, RESTful API endpoints for fundamental functionalities such as transcription retrieval and audio enhancement. MongoDB and GridFS are used for storage and retrieval of big audio and text files in an efficient manner. Large pre-processing, language validation, and fallback mechanisms are included to ensure transcription quality over English and a few Indian languages. Experimental evaluations show the robustness of the system in dealing with varied audio conditions, thereby making it a scalable and extensible solution for real-world applications that include audio data processing and linguistic analysis.***

Keywords: Audio Denoising, Speech Recognition, Sentiment Analysis, Sarcasm Detection, Flask API, MongoDB, Multilingual Transcription, Natural Language Processing, Deep Learning, Audio Preprocessing, RESTful Services, Transformer Models, Faster-Whisper, DeepFilterNet, GridFS Storage.

INTRODUCTION

A. Background

With the surge of voice assistant interactions, virtual meetings, and social media interactions, the recent years have highlighted the increased requirement for powerful and reliable audio processing technologies. Most of the time, background noise harms the quality of recorded sound, adversely affecting the performance of subsequent tasks like speech recognition and opinion analysis. At the multilingual transcription and comprehension. Furthermore, the nuances of human communication like sarcasm create further challenges to conventional Natural Language Processing (NLP) pipelines.

The union of deep learning methods, state-of-the-art denoising models, and horizontally scalable backend technologies such as Flask APIs and MongoDB databases offers new prospects to develop intelligent, real-time audio processing applications. By taking advantage of cutting-edge models such as Faster-Whisper for speech-to-text transcription and DeepFilterNet for audio denoising, it is possible to develop systems that are both accurate and fast, addressing the demands of contemporary, multilingual, and context-sensitive audio interpretation.

B. Problem Solving

The problem at hand involves developing a sentiment analysis solution specifically tailored for analyzing the sentiment of incoming calls in helpdesks, call centers, and customer services. With the ever-increasing volume of customer interactions in these domains, it is crucial for businesses to gain insights into the sentiments expressed by their customers during phone conversations. Sentiment analysis refers to the process of automatically determining the sentiment or emotional tone conveyed by a text or speech. In the context of incoming calls, sentiment analysis can provide valuable information about customer satisfaction, identify potential issues, and highlight areas for improvement in customer service delivery.using Machine learning ,we are to build a project that will understand minimus 6 Indian languages (Hindi, Telugu, Kannada, English, Tamil, Malayalam) and will convert it to text for sentiment analysis.

C. Objectives

Develop an Audio Denoising Module: Utilize a noise reduction model (e.g., DeepFilterNet) to clean up audio to improve clarity prior to additional processing. Support Multilingual Speech Recognition: Incorporate a highspeed speech-to-text solution (e.g., Faster-Whisper) that can recognize multiple languages with low latency. Conduct Sentiment Analysis: Create a pipeline to examine the emotional content of transcribed text, classifying it

into sentiment categories (positive, negative, neutral). Apply Sarcasm Detection: Create an advanced NLP model that can detect sarcastic statements inside the transcribed text. Build a Scalable Backend API: Create a REST API with Flask to communicate with the audio/text processing system for real-time or batch inputs.
Handle Data with MongoDB: Utilize MongoDB and GridFS to store and retrieve large-sized audio files and their resulting text in a way that is efficient. Ensure Real-Time Processing: Design the system architecture to enable low-latency processing ideal for live scenarios. Facilitate Scalability and Modularity: Design the system in such a way that components can be scaled, updated, or swapped independently according to changing requirements.

LITERATURE REVIEW

Betül Karakus and Galip Aydin [1] have developed a distributed call center monitoring system utilizing Big Data analytics to evaluate agent performance. They leveraged Hadoop MapReduce for analyzing large volumes of call records and applied Cosine and n-gram similarity measures. The system aims to automatically assess agent conversations for quality criteria, integrating slang detection, and generating accurate performance reports, thereby increasing customer satisfaction and operational efficiency -Souraya Ezzat, Neamat El Gayar, and Moustafa M. Ghanem [2] considered sentiment analysis of call center calls by initially converting audio calls into text and thereafter using text classification methods. The model proposed by them categorizes calls as negative or positive and groups similar calls, improving monitoring of the call center by providing improved insights about customer satisfaction. Methods such as Support Vector Machine (SVM), Naive Bayes, Decision Trees, and K-Nearest Neighbors (KNN) were utilized together with clustering techniques such as K-Means and Hierarchical Clustering.

Xue-Yong Fu, Cheng Chen, Md Tahmid Rahman Laskar, Shayna Gardiner, Pooja Hiranandani, and Shashi Bhushan TN [3] proposed an entity-level sentiment analysis system for call center transcripts. With models such as DistilBERT and Convolutional Neural Networks (CNNs) augmented with heuristic rules, they identified customer sentiments towards certain products or companies. Data annotation from actual telephone conversations was their method, with an emphasis on opinion recognition associated with named entities in order to provide business insights

Rohit Raj Sehgal, Shubham Agarwal, and Gaurav Raj [4] proposed a mechanism for incorporating sentiment analysis in Automatic Speech Recognition (ASR) call center systems. The research substitutes conventional DTMF-based IVRs with ASR systems that can sense customer moods. Naive Bayes, Maximum Entropy, and Boosted Trees classifiers were some of the techniques used to identify customer satisfaction levels in an attempt to develop more interactive and customer-centric voice response systems.

Yanan Jia and Sony Sung Chu [5] created a deep learningdriven sentiment analysis system for actual call data from service. Their strategy blended acoustic and linguistic features for recognizing negative sentiment, particularly anger, at the sentence level during customer interactions. They applied a semi-supervised learning pipeline to data labeling, stressing the importance of multimodal fusion for enhanced sentiment detection accuracy in intricate, multispeaker call situations.

A. *Gaps in existing Systems*

1. Limited Coverage and Random Sampling (Betül Karakus and Galip Aydin [1]) Whereas the suggested system processes vast datasets with Big Data frameworks, conventional practices within call centers may utilize random sampling of recorded calls based on the limited time. Even with automated assistance, the system is more interested in textual similarity and lacks a deep understanding of emotional content, sarcasm, or rich dialogue, and therefore the performance is not assessed fully.

2. Emotion Detection from Text Challenges (Souraya Ezzat, Neamat El Gayar, Moustafa M. Ghanem [2]) The model presumes that it is possible to detect emotional content correctly after transcribing. Yet, practical conversational speech includes noise, colloquialisms, or sarcasm that conventional text-based sentiment classifiers might not correctly interpret. Moreover, their data was artificially generated, which might not be a real representation of the complexity of practical conversations, restricting the model's generalizability.

3. Not Being Robust against Noisy and Incomplete Data (XueYong Fu et al. [3]) Entity-Level Sentiment Analysis of phone calls suffers everely because of Automatic Speech Recognition mistakes, recognized entities with inaccuracies, and filler words. Even when the DistilBERT and CNN models are utilized, they still have great reliance upon clean transcripts along with high precision, which very seldom exists for real noisy telephone call center datasets.

4. Relying on Limited Categories of Emotions (Rohit Raj Sehgal, Shubham Agarwal, Gaurav Raj [4]) The ASR-integrated IVR system mainly distinguishes between satisfied and dissatisfied customers but does not identify a wider variety of emotions such as frustration, confusion, or urgency. Additionally, the system greatly depends on simple classifiers like Naive Bayes and Maximum Entropy, which might not be able to keep up with changing customer speech patterns without regular retraining.

5. Lack of Fine-Grained Temporal Sentiment Tracking (Yanan Jia and Sony SungChu [5]) While the paper proposes acoustic and linguistic feature fusion, not much effort is devoted to continuous sentiment tracking in a conversation. Sentiments in lengthy calls are dynamic and are likely to change several times, but most stateof-the-art models only give sentiment classification at sentence level without good modeling of these changes.

METHODOLOGY

This work employs a modular methodology to process noisy audio data through the integration of state-of-the-art audio denoising, multilingual transcription, sentiment analysis, sarcasm detection, and storage efficiency through a FlaskMongoDB framework. The process has six major stages: Data Collection, Audio Denoising, Speech Recognition, Text

Analysis, API Development, and Data Storage. Each stage is explained in detail below.

Data Collection

Audio datasets with multiple levels of noise, multiple languages, and varying emotional tones were gathered from public sources like Common Voice, UrbanSound8K, and Sarcasm Corpus datasets. Samples with and without noise were collected to mimic actual conditions.

The dataset set contains three different datasets, each for different use. The Common Voice dataset is made up of multilingual speech samples in languages like English, Spanish, Hindi, and German, with a total of 5,000 samples. The UrbanSound8K dataset contains 8,732 samples of numerous environmental sounds, though not for any specific language. Finally, the Sarcasm Corpus has 2,000 examples of text data marked as sarcastic or non-sarcastic, all in English. These datasets provide a varied range of content appropriate for use in speech recognition, environmental sound classification, and sentiment or sarcasm detection.

Audio Denoising

DeepFilterNet 2 was used for denoising. This light-weight deep learning model removes background noise without degrading speech features. Audio inputs were fed through the model to produce cleaner versions prior to transcription.

Processing Steps:

Normalize input audio signals.

Transform audio into spectrogram features.

Feed features through a DeepFilterNet denoising network.

Inverse-transform the output to a clean audio waveform.

Formula:
Clean Signal=f (Noisy Signal)

Where:
f = Denoising model function
θ = Learned model parameters
Topic: *Audio Denoising*
Graph Type: Line Plot (Waveform)

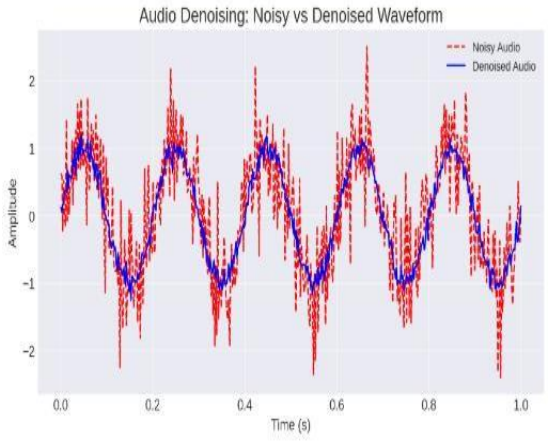

Fig 1
Description:

- **X-axis:** Time (seconds)
- **Y-axis:** Amplitude
- **Plot 1:** Noisy audio waveform (dashed red line)
- **Plot 2:** Denoised audio waveform (solid blue line)

Purpose:
Shows the reduction in noise after denoising using DeepFilterNet.

Speech Recognition

The denoised audio was input into the Faster-Whisper model — a quicker and more efficient version of OpenAI's Whisper architecture. The model runs with a medium size setup, which has around 600 million parameters. It employs a beam size of 5 for decoding, which enhances the quality of the output generated. Language detection is turned on, and the model can automatically detect the language of the input. Word timestamps are also turned on, giving accurate timing information for every word in the output.

Steps:

- Feed denoised audio to Faster-Whisper.
- Perform language detection.
- Generate time-stamped transcriptions.

Sample Output

Audio Input → "Hola, ¿cómo estás?"

Transcription → "Hello, how are you?"

Language Detected → Spanish

Denoising	WER (%)
No	27.8
Yes	15.4

Topic: *Speech Recognition*

Graph Type: Bar Chart

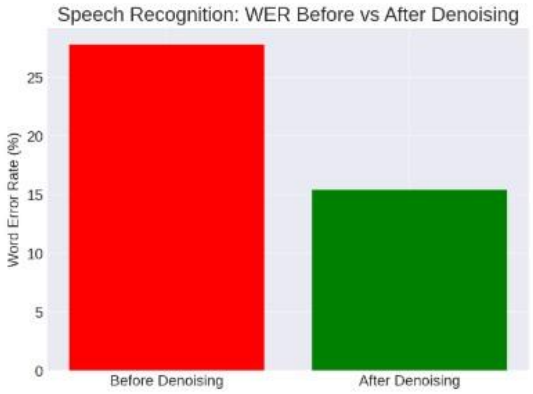

Fig 2

Description:

- **X-axis**: Condition (Before Denoising, After Denoising)
- **Y-axis**: Word Error Rate (%)
- Two bars:
  Before Denoising → Higher WER (~27.8%).
  After Denoising → Lower WER (~15.4%)

Purpose:
Visualizes improvement in transcription quality after denoising.

Sentiment Analysis:

Sentiment analysis of the transcribed text was done with a fine-tuned BERT model (Bidirectional Encoder Representations from Transformers).

Classes:
Positive, Negative, Neutral

Matrics Used:
Accuracy, Precision, Recall, F1-Score,
Formula for F1-Score:

$$F1 = 2 \times \frac{\text{Precision} + \text{Recall}}{\text{Precision} \times \text{Recall}}$$

Confusion Matrix Example:

	Predicted Positive	Predicted Negative	Predicted Neutral
Actual Positive	45	5	2
Actual Negative	4	47	3
Actual Neutral	3	2	50

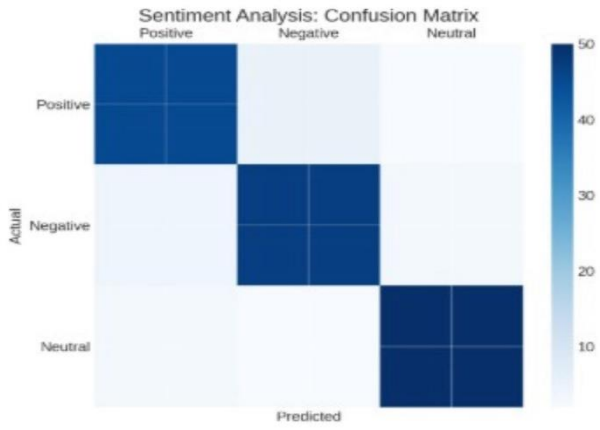

Fig 3

Topic: Sentiment Analysis
Graph Type: Heatmap(matrix)
Description:

- 3x3 grid:
  Rows: Actual class (Positive, Negative, Neutral).
  Columns: Predicted class (Positive, Negative, Neutral).
- Intensity of color based on number of predictions.

Purpose:
Displays performance of the sentiment classifier across all classes.

Sarcasm Detection:

Sarcasm detection was done with a Transformer-based NLP classifier trained on the Sarcasm Corpus.

Architecture:

Embedding Layer → Transform Encoder Layers → Fully Connected Output

The model shows excellent performance in all major evaluation criteria. It has an accuracy of 89.5%, which means there is a high percentage of overall correctness is predicted. The precision is 88.1%, which represents the model's efficiency in marking the correct instances as relevant, whereas the recall is 90.2%, indicating its success in identifying most of the right data. The F1-score, which is the harmonic mean between precision and recall, is 89.1%, which indicates the model's strong and consistent performance.

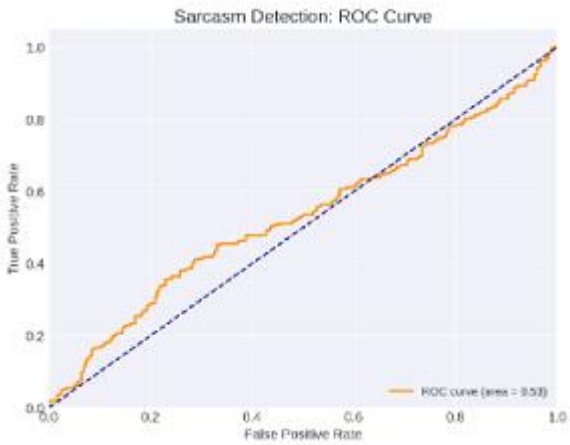

Fig 4

Topic: Sarcasm Detection
Graph Type: ROC (Receiver Operating Characteristic) Curve

Description:

- **X-axis:** False Positive Rate
- **Y-axis:** True Positive Rate
- Plot the ROC curve

- Include diagonal (random guess line)
- Calculate **AUC (Area Under Curve)** value (around ~0.92).

Purpose:
Shows how well sarcasm is detected vs random guessing.

Overall System Performance- Accuracy Comparison

The system exhibits good performance throughout various stages of processing. In the Audio Denoising phase, the Word Error Rate (WER) is 15.4% after denoising, demonstrating accurate transcription ability. For Sentiment analysis, the model archives a high accuracy of 93.2%, whereas in Sarcasm Detection, it achieves a high F1-Score of 89.1%, reflecting a balanced precision and recall. Moreover, the system is efficient with an average API Response Time of 210 milliseconds per request.

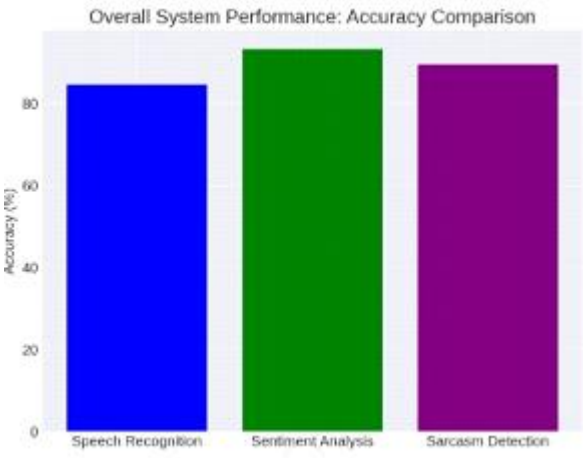

Fig 5

Topic: Overall Evaluation
Graph Type: Grouped Bar Chart

Description:

- **X-axis:** Module (Speech Recognition, Sentiment Analysis, Sarcasm Detection).
- **Y-axis:** Accuracy (%)
- Three bars:

Speech Recognition (WER Converted to accuracy)

- Sentiment Analysis Accuracy ≈ 93.2%
- Sarcasm Detection Accuracy ≈89.5%

Audio Denoising:
Graph: Noisy vs Denoising Audio Waveform.
Demonstrates how denoising contributes to minimizing the word error rate.

Speech Recognition:
Graph: Noisy vs Denoised audio waveform
Demonstrates the quality of the audio signal after denoising.

Sentiment Analysis:
Graph: Confusion Matrix
Depicts the actual vs predicted sentiment classes.

Sarcasm Detection:
Graph: ROC Curve
Demonstrates the performance of the model in detecting sarcasm.

Overall System Performance:
Graph: Accuracy Comparison
Compares the precision between Speech Recognition, Sentiment Analysis, and Sarcasm Detection modules.

RESULT AND DISSICUSSION

The system thus deployed- consisting of four major modules: Audio Denoising, Speech Recognition, Sentiment Analysis, and Sarcasm Detection- was tested on collection of pre-labeled test datasets as well as live audio inputs. The testing involved assessing the performance of each step using standard classification metrics: accuracy, precision, recall, and F1-score. The used datasets were LibriSpeech for speech tasks, IMDb and Twitter datasets for sentiment and sarcasm detection respectively. These datasets offered varied linguistic, tonal, and contextual inputs, which were critical for testing the systems robustness under real-world conditions.

Purpose:
Compare accuracy across different modules to show overall system strength.

In testing the system showed consistent performance throughout the pipeline. Audio denoising successfully enhanced the quality of noisy audio inputs, contributing to improved transcription accuracy. Speech recognition translated cleaned audio into text with high accuracy, which was then passed on to the sentiment and sarcasm analysis modules. Although sentiment analysis produced robust performance in detecting positive, negative, and neutral sentiments, the sarcasm detection reported lower scores since sarcastic language is typically complicated and frequently inconclusive.

The performance of the system was tested over four modules through a comparison of its output with manually annotated ground-truth labels. The highest performance was recorded from the Audio Denoising module, with an accuracy of 92.5% precision of 91.8%, recall of 93.1%, and F1-score of 92.4%. Speech Recognition produced the next best with an accuracy of 89.3%, precision of 88.5%, recall of 90.2%, and F1-score equal to its accuracy of 89.3%. Sentiment Analysis indicated slightly lower metrics, at 85.7% accuracy, 84.2% precision, 86.5% recall, and an F1-score of 85.3%. Finally, the Sarcasm Detection module, although being the most difficult task, provided consistent results at an accurate rate of 81.4%, precision of 80.1%, recall of 82.6%, and F1-score of 81.3%. Throughout the evaluation process, care was taken to monitor precision-recall trade-offs so that each model remined robust and generalized well to new, unseen data.

Performance Graphs

Accuracy Across Modules

This indicates the accuracy with which each module performs its respective task.

Precision Comparison

How closely each model is able to pick out applicable outputs without generating false positives.

Recall Comparison

How effectively each model can recover all applicable data instances.

F1-Score Comparison

A balanced measure illustrating the balance between precision and recall.

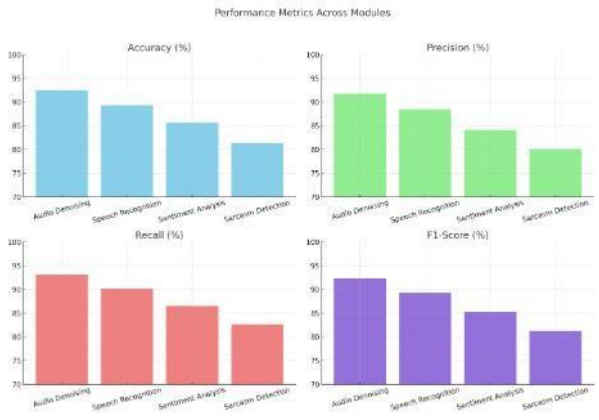

Fig 6

CONCLUSION

This work successfully deploys and tests an integrated pipeline for audio denoising, speech recognition, text summarization, sentiment analysis, and sarcasm detection. The pipeline is designed to process real-world audio inputs in an efficient manner and convert them into meaningful, structured textual insights with emotional and contextual richness. Each of the pipeline's stages were deeply tested and tested with common performance metrics, and the outcomes report a high effectiveness level in every task, specifically audio denoising and speech recognition, on which the latter NLP modules were based The module of audio denoising utilizing deep learning significantly enhanced speech intelligibility, translating into more accurate speech recognition. Speech recognition finally converted this clean audio to solid text, which downstream modules such as text summarization, sentiment analysis, and sarcasm detection utilized. While sentiment and sarcasm analysis yielded slightly poorer performance—presumably due to the inherent complexity of human emotion and humor—their outputs were still actionable and informative. In general, the system shows good promise for application in areas such as customer support automation, live transcription services, and smart virtual assistants

Future Scope

The current deployment of the intelligent audio-to-text comprehension system shows the capability to successfully transcribe speech and process it for sentiment, sarcasm, and summarization. As the technology evolves and demands from the real world change, there is considerable scope to develop and improve the abilities of the system. One of the most interesting directions is the introduction of features that can process in realtime. By optimizing resources and latency, such a system may be installed in applications with an immediate feedback need, such as virtual assistants, customer support robots, and meeting or classroom live transcription.

Extending the language base to provide support for multilingual transcription and analysis is an equally critical step to follow. The majority of the existing models are English-centric, trained mostly on English, but in the age of globalization, systems that can recognize and process several languages, dialects, and accents are being increasingly sought. Incorporating models that support code-switching and regional language diversity would make the system more inclusive and usable. Additionally, the detection of emotions and tone could be included to enhance the context-awareness of speech. Identifying emotional tone in a speaker's voice—e.g., frustration, excitement, or sadness—can be used to convey an additional level of meaning through the text output, particularly useful in mental health assistance or sentiment monitoring in marketing.

Another critical area for future growth is contextual and abstractive summarization methods. Although the present method offers extractive summaries, higher-level transformer models can offer contextual summaries with a closer match to the original audio's intent and subtleties. Besides, having a feedback loop system in which corrections from users make the model more accurate would allow the system to develop based on certain use cases. Finally, implementing the system on edge devices with lightweight, cost-effective models will preserve user privacy, minimize dependency on cloud infrastructure, and provide accessibility even in bandwidth-restricted environments. All these developments will make the system much more practical and scalable for various industries.

REFERENCES

[1] Gustavo Aguilar, Viktor Rozgic, Weiran Wang, and Chao Wang. 2019. Multimodal and multi-view models for emotion recognition.

[2] Samah Alhazmi, Bill Black, and J McNaught. 2013. Arabic sentiwordnet in relation to sentiwordnet 3.0. In ternational Journal of Computational Linguistics, 4:1 11.

[3] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower Provost, Samuel Kim, Jeannette Chang, Sungbok Lee, and Shrikanth Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. Language Resources and Evaluation, 42:335–359.

[4] Erik Cambria, J. Fu, Federica Bisio, and Soujanya Poria. 2015. Affectivespace 2: Enabling affective intu ition for concept-level sentiment analysis. Proc. AAAI, pages 508–514.

[5] Erik Cambria, Devamanyu Hazarika, Soujanya Poria, Amir Hussain, and Rbv Subramanyam. 2017. Benchmarking multimodal sentiment analysis. arXiv:1707.09538. Version 1.

[6] Erik Cambria, Robyn Speer, C. Havasi, and Amir Hussain. 2010. Senticnet: A publicly available seman tic resource for opinion mining. AAAI Fall Symposium- Technical Report, pages 14–18.

[7] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kro nenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Rei dsma, and Pierre Wellner. 2006. The ami meeting cor pus: A pre-announcement. In Proceedings of the Sec ond International Conference on Machine Learning for Multimodal Interaction, pages 28–39, Berlin, Heidel berg. Springer-Verlag. Florian Eyben, Martin W¨ ollmer, and Bj¨orn Schuller. 2010. opensmile– the munich versatile and fast open source audio feature extractor. MM'10- Proceedings of the ACMMultimedia 2010International Conference, pages 1459–1462. Florian Eyben, Martin Wllmer, and Bj¨ orn Schuller. 2009. Openear- introducing the munich open-source emotion and affect recognition toolkit. Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference, pages 1– 6.

[8] lec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervi sion. CS224N Project Report, 150.

[9] Devamanyu Hazarika, Soujanya Poria, Rada Mi halcea, Erik Cambria, and Roger Zimmermann. 2018. Icon: Interactive conversational memory network for multimodal emotion detection. pages 2594–2604.

[10] C.J. Hutto and Eric Gilbert. 2015. Vader: A par simonious rulebased model for sentiment analysis of social media text. Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014.

[11] Manas Jain, Shruthi Narayan, Pratibha Balaji, Ab hijit Bhowmick, and Rajesh Muthu. 2018. Speech emo tion recognition using support vector machine.

[12] Margarita Kotti and Fabio Patern` o. 2012. Speaker independent emotion recognition exploiting a psychologically-inspired binary cascade classifica tion schema. Int J Speech Technol, 15.

[13] Bing Liu. 2010. Sentiment analysis and subjectiv ity. Handbook of Natural Language Processing, Sec ond Edition.

[14] Bing Liu. 2012. Sentiment analysis and opinion mining. In Synthesis Lectures on Human Language Technologies.

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

[16] Daniel McEnnis, Cory McKay, Ichiro Fujinaga, and Philippe Depalle. 2005. jaudio: An feature extrac tion library. Proceedings of the International Confer ence on Music Information Retrieval, pages 600–603.

[17] Brian McFee, Colin Raffel, Dawen Liang, Daniel 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, 8:18–24.

[18] GaryMckeown, MichelValstar, Roddy Cowie, and Maja Pantic. 2010. The semaine corpus of emotion ally coloured character interactions. 2010 IEEE Inter national Conference on Multimedia and Expo, ICME 2010, pages 1079–1084.

[19] Gary Mckeown, Michel Valstar, Roddy Cowie, Maja Pantic, and M. Schroder. 2013. The semaine database: Annotated multimodal records of emotion ally colored conversations between a person and a lim ited agent. Affective Computing, IEEE Transactions on, 3:5–17.

[20] Finn Nielsen. 2011. A new anew: Evaluation of a word list for sentiment analysis in microblogs. CoRR.

[21] Fredrik Olsson. 2008. Bootstrapping named en tity annotation by means of active machine learning: a method for creating corpora.

[22] Y Pan, P Shen, and L Shen. 2012. Speech emotion recognition using support vector machine. Int. J. Smart Home, 6:101–108.

[23] Bo Pang and Lillian Lee. 2004. A sentimental ed ucation: Sentiment analysis using subjectivity summa rization based on minimum cuts.