Deep Learning Applications

TITLE: Sentiment Analysis of Audio files using BERT (Bidirectional Encoder Representations from Transformers)

NAME: ANJU MULLAKKARAMALIL SUKUMARAN NAIR

STUDENT ID: MUL23618197

Table of Contents

1. Background and Problem Definition

The ability to analyze sentiments from audio input has broad applications across diverse industries. From customer feedback analysis to monitoring mental health, understanding the emotional tone behind spoken language offers valuable insights into user intentions and emotions.This brings value in major areas, including customer service and mental health: it can help detect dissatisfaction early on, thus enabling timely intervention. Secondly, in mental health, the trace of distress in conversations can be identified, therefore helping professionals in the discharge of better care.

These are despite the fact that traditional methods of carrying out sentiment analysis usually rely on text-based inputs, with most not considering the intelligently nuanced information embedded in audio formats. Audio data encloses more key information than just words through intonation, pitch, and pauses to depict the emotional condition of the speaker. This work bridges this gap by designing a real-time sentiment analysis system using state-of-the-art speech-to-text transcription and sentiment classification.

Audio data processing has its own issues. Speech recognition has to consider background noise, different accents, speed of talking-accurately transcribing text. In sentiment analysis, capturing subtlety and complexity around emotions often couch potato in their presence, like sarcasm, irony, or ambiguity, which is usually tough to decide by models. This work automatizes the whole workflow by integrating OpenAI's Whisper for robust speech recognition with BERT for reliable sentiment classification, making audio sentiment analysis accessible and efficient. The system is intuitively designed to be usable by non-technical users, democratizing the ability to glean actionable insights from spoken language in real time.

2. Proposed Deep Learning Solution

The following solution to the challenge described will complete the task of real-time audio sentiment analysis by using two leading-edge deep learning models. They are as follows:

A Robust Speech-to-Text Transcription Model: Whisper, developed by OpenAI, is the state-of-the-art model on a wide variety of languages, accents, and audio conditions. It can give very good transcription accuracy and is designed to work even in environments with heavy background noise. Whisper will work fine with everything from poor recording conditions to poor input audio quality, as it draws on a large multilingual and multitask dataset. Unlike the traditional systems, Whisper's complex neural network architecture gives flexibility and robustness when processing complex speech input.

BERT (Bidirectional Encoder Representations from Transformers): BERT is a pre-trained transformer-based model which understands syntax, semantics, and the context of linguistics. It has been specifically developed for the purpose of sentiment categorization; hence, BERT is very strong in the investigation of emotional tone in sentences. It has the added capability to pick up even minor emotional cues within transcription data, thanks to the bidirectional mechanism it uses that can comprehend the whole context of a sentence.

These models are jointly deployed in this user interface created on **Streamlit.** The system functions by first recording audio, transcription of audio content using Whisper, and further uses BERT for sentiment analysis on this transcription. Accordingly, it detects the emotions such as positive, negative, and neutral which helps customers create actionable information extracted from spoken words. Thus, these integrations allow the proposed system to be not only efficient but user-friendly to address advanced-level sentiment analysis using spoken languages in a very general way.

3. Data Selection

This involves basically two datasets on which the model relies to give correct and efficient end-to-end sentiment analysis in the following order:

1. Audio Data: The first tier raw input will be real-time users' audio. The data will be embedded by using Whisper, which has been pretrained on a big multilingual, multitask dataset. The current dataset encompasses a wide range of languages, accents, and speech conditions that enable the model to achieve good transcription results on a wide range of datasets without any further fine-tuning. Whisper is pretty robust: it can help get a transcription even from audios that have a lot of background noise or multiple speaker styles. Applying such an advanced architecture enables raw audio to be turned into high-quality text that then forms the basis for subsequent sentiment analysis.

2. Sentiment Data For sentiment classification, the system accepts a labeled dataset saved in a file called labels.csv. This includes samples of texts that fall into various sentiments such as positive, negative, and neutral. This will then act as the ground truth for fine-tuning the BERT model so that transcription outputs can be mapped to appropriate sentiment categories. Each labelled text example is representative for different emotional expressions to be covered as well as varied contexts by the model.

Much care was taken in the selection and preparation of a very diverse dataset for fine-tuning BERT to improve its performance. Representatives of various linguistic and emotional nuances can be found therein, hence enabling the model to tackle such intricate scenarios: ambiguity in statements, subtle emotional expressions, among others. The diversity of data is highly essential for accuracy and reliability in different applications that may arise for sentiment analysis.

4. Preparing the Data

The following are the key pre-processings required both for compatibility and optimization. Most of them include audio and text. In fact, raw input must go through several steps to reach this format pertinent to training or inference for good, valid, and reliable analyses.

Audio Preprocessing Recording: The sounddevice library is a very powerful library used for recording audio from the user's microphone. In the recording, a sample rate of 16 kHz was selected because it is a good compromise between computational efficiency and high-quality capture of audio. This sample rate also agrees with the Whisper model since it requires even input for transcription.

Normalization: These signals were then further adjusted to allow for changes in the input audio levels. This step normalizes the volume of the audio and ensures that no recording is too loud or too quiet, which would affect the correctness of the transcription. Normalization is very important in the case of processing audio from various environments and devices; it reduces the difference due to different microphone sensitivity or user distances.

Audio Format Conversion: The audio recordings were stored in 16-bit PCM WAV format, which is a very common format that can easily support Whisper model performance. Additionally, this conversion includes resampling, assurance of bit depth, and channel count, further playing an important role in the model performance.

Text Preprocessing

Tokenization: The transcribed text from the Whisper model was tokenized using BERT\u2019s tokenizer. This step splits the text into subword units and adds special tokens, such as [CLS] and [SEP], to mark the beginning and end of the input sequence. Even complicated or unclear sentences are tokenized by BERT in such a way that it retains the contextual links of the words involved.

Label Encoding: Sentiment labels were encoded numerically using the LabelEncoder from the Scikit-learn library. For example, the numerical values 0, 1, and 2 represent positive, negative, and neutral sentiments, respectively. Label encoding will convert the categorical data into a numerical representation that can be fed into the classification head of the model. In turn, label encoding ensures consistency in matching text inputs with the correct sentiment classes throughout both training and testing.

These preprocessing methods finally converted the audio and text data into structured, model-ready representations. Furthermore, it was important that the integrity of the data be preserved so that models could work effectively and accurately during both training and inference.

5. Defining the Deep-Learning Model

Architecture and Design Decisions for the BERT Model

BERT model stands for Bidirectional Encoder Representations from Transformers and acts as one of the benchmark models for performing tasks that relate to natural language processing. Its architecture represents deep linguistic linkages and environmental nuances through the incorporation of the self-attention mechanism.

Key Attributes:

Transformer Architecture: Since the meanings of words heavily relied on contexts for tasks such as sentiment analysis, BERT applied a multi-layer transformer architecture with self-attention mechanisms to record links between the words of the sentences.

Bidirectional Context Understanding: Whereas previous models were designed to process the text in flow, BERT reads phrases in both directions. This makes it very successful in identifying delicate sentiments in text by catching the contexts of both its preceding and succeeding words.

Fine-tuning: Even though pre-trained on a massive general corpus such as Wikipedia and BooksCorpus, in this project, the BERT is fine-tuned on sentiment-labeled datasets. Fine-tuning changes the weights of the model to specialize in sentiment classification so that there can be accurate detection of positive, negative, and neutral sentiments.

The Classification Mechanism in BERT involves a dense layer followed by softmax to make appropriate outputs. Outputting probabilities over each sentiment class, the appropriate label may be assigned from the highest of these probabilities.

The system combines the Whisper model for transcription with BERT for sentiment analysis; by doing so, it capitalizes on the strengths of each architecture. Where Whisper adds robustness for accurate text output, BERT's understanding of the context can spot even the most complex sentiment variations. Therefore, these design choices collectively provide a robust solution for real-time audio sentiment analysis.

6. Training and Fine-Tuning Model

Fine-tuning Process of BERT

The pre-trained language understanding capability of the BERT model was fine-tuned in this project to perform sentiment classification. It is fine-tuned so that it gets tailored for text classification, specifically categorizing into positive, negative, and neutral sentiment classes.

1. Preparing the dataset

The fine-tuning dataset was the information in the labels.csv file. This file contains samples of labeled text, each having a class which corresponds to sentiment. To ensure that during the training phase, bias would not occur; it was ensured that all feelings were almost equally represented in the dataset. For compatibility with the model, the text labels needed to be changed into numerical values using Scikit-learn's LabelEncoder.

2. The Instructional Procedure

For BERT to operate optimally, some key hyperparameters had to be tuned. Some selected ones are enumerated below.Learning Rate: 2e-5 is kept to its value as it will help in learning optimally for the model with extremely minimal chance of overshooting onto ideal weight changes.

Batch Size: A batch size of 16 has been chosen because it strikes a good balance between computational cost efficiency and gradient estimation accuracy.

Epochs: It converges in three epochs without overfitting.

Optimizer: The optimization algorithm used is AdamW. This is optimized to train deep learning models and it's a version of the Adam Optimizer. It is implemented in order to avoid overfitting in a model by utilizing the weight decay technique for regularization purposes. Besides that, linear learning rate decay was used to gradually decrease the learning rate during training and hence enhance stability and convergence.

3. Challenges

One major problem with training was overfitting. In the case of the current work, it is avoided with the help of a validation set, which ensures that performance will be good enough for unseen data. The method of dropout regularization is used; some neurons are randomly turned off during training in order not to rely on too specific features of the model.

Usage of Whisper for Transcription

Whisper is a speech-to-text model that has robustness and can work directly without extra training. It supports everything out-of-the-box for handling noisy conditions, different accents, and speaking speeds with the pre-trained state. The result coming from Whisper will be very fine, directly from the input audio and it forms a very important ground in the entire sentiment analysis workflow. Its advanced architecture, combined with the multilingual training dataset, gives it the strength not to need any further tuning to work efficiently in this system.

7. Testing Model with New Data

This test of the model's efficiency in a different data set was necessary to ascertain its reliability and robustness. In testing, the conditions were varied to precisely determine the efficiency of both the transcription and the classification of sentiments. The main performance metrics are the F1 Score for the sentiment categorization and the WER for transcription accuracy.

1. Transcription Accuracy

The WER metric was utilized to assess performance for the Whisper model. This metric computes the ratio between the errors-insertions, deletions, and substitutions-to the total number of words in the ground truth transcription. When clear audio inputs are processed, the model showed very high accuracy with very minor and negligible errors, while in a noisy environment or with speech overlap, transcription errors occasionally occur. Most were mispronunciations of words whose phonemes resembled each other, or omission of phrases pronounced too soft. Whisper's noise-resilient architecture however produced intelligible transcriptions to be used on sentiment analysis applications.

2. Classification of Sentiment Precision

Then, the sentiment analysis was measured using the F1 Score that balances recall with precision.

The BERT model proved to work reliably in the classification of straightforward sentiments, such as clearly positive or negative expressions. For instance, "This is excellent" was well classified as positive, while, for example, sarcasms and/or ambiguous sentences whose sentiment was implicit required more insight from the model. These findings may suggest that further fine-tuning could be necessary in diverse datasets to improve this model's performance under such complex language usage.

Testing Conditions

Testing was done on the basis of pristine recordings and noisy, environmentally interfering audio inputs. Accuracy was checked by comparing the sentiment predictions of the algorithm against the ground truth labels. The presented deep analysis has shown how good the system performs and also pointed out the areas where the system needs further improvements, especially for those cases with complicated emotions.

8. Deploying Model

System Installation Streamlit is a web framework in Python that allows building interactive web applications. This was the framework under which the system was installed to provide real-time performance, efficiency in the system, and accessibility for users. The steps followed in transforming the sentiment analysis system into a working website are the following:

1. Environment Configuration

Initially, Python 3.8 or later was installed to guarantee compatibility with all the needed libraries and frameworks, and then the environment was prepared for application deployment.The authors created a virtual environment to separate these libraries so that they do not conflict with other projects. Requirements.txt also included a list of all the required libraries, including SoundDevice, Transformers, and Streamlit, making the installation easier and reproducible because of the use of a single command: pip install -r requirements.txt.

2. Workflow for Applications

The workflow of the deployed application is pretty lucid and clear in its setup flow:

Record Audio: The user may record audio input from a microphone without jerks using the Streamlit GUI integrated with a library called SoundDevice.The script saves a WAV file of the audio recorded for processing.

Transcription: Audio recording is loaded into the Whisper model, which ensures that material is correctly transcribed.

Sentiment Analysis: The fine-tuned BERT model further categorizes the transcribed text into three sentiments: neutral, positive, and negative. These results are then displayed in real time on the web interface.

3. Optimization

Following are some optimizations that were made to enhance the user experience and responsiveness of the applications:

Model Preloading: Preloading Whisper and BERT models while the application initializes reduces latency in later uses since there is no need to reload models with each request.

GPU Acceleration: GPU support was enabled if it was available to help speed up the inference, especially for transcription and sentiment analysis activities. The longer the audio input or in the event of concurrency in users' interactions, the faster it will process.

The different deployment procedures at work resulted in an application that is stable, intuitive in nature, and provided real-time sentiment analysis for audio inputs through any modern browser.

9. Results and Analysis

The system accomplished the project objective by delivering a robust and user-friendly real-time sentiment analysis solution, as is evident from its very good performance on transcription and reasonably good performance on sentiment categorization.

Transcription Performance

It is observed that when the input audio is clear and of good quality, the Whisper model achieves near-perfect transcription accuracy. Even in cases where there is moderate noise, its noise resilience sees it maintain coherence with perhaps just a few minor errors. An example is the spoken line "I am happy today," which it transcribed perfectly and accurately. While minimal errors occurred for instances when it was a bit noisier in the background and articulation could sometimes not be clear.

Sentiment Analysis

BERT did well and was consistent when classifying feelings as positive, negative, or neutral in cases where the inputs were candidly simple in nature. Transcription like "This is amazing" and "I disliked the experience" got classified correctly as positive and negative sentiment, respectively. However, this model has problems whenever ambiguous or subtle languages are used sarcasm and irony among them. Such set phrases, like "Oh great, another delay", were at times misclassified; that suggests the model does not understand sentiments hidden behind complex or subtle contexts.

Overall Analysis

The results show it works very well for typical use cases but may be further improved by fine-tuning the BERT model on datasets containing diverse emotional nuances; advanced noise-cancellation techniques for Whisper would further improve transcription accuracy in challenging audio environments. Such a finding provides a sound basis for future refinement and wider applicability.

Challenges:

1. Noisy Environments: Reduced transcription accuracy.

o Solution: Implement noise-cancellation preprocessing.

2. Ambiguous Sentiments: Difficulty with sarcasm or complex emotions.

o Solution: Fine-tune BERT with more diverse datasets.

10. Conclusion

The presented project succeeded in demonstrating the integration of Whisper and BERT models with state-of-the-art deep learning techniques for effectively bridging the gap from audio processing to actionable insights and has been able to deliver real-time sentiment analysis using only audio as input, making the system versatile for various applications. It will make it easier for industries like customer care, social media monitoring, and mental health diagnosis to get more valuable information about emotions and contexts from speech. This intuitive design allows all users to apply it, irrespective of the user's technical background; thus, increasing the coverage area.

Key Highlights

Several critical observations were made during the development and testing phases of the project:

- **High-Quality Audio:** The quality of the input audio affects transcription accuracy to a great degree. Cleaner recordings equate to more accurate transcriptions; hence, it is very important to use high-quality microphones or preprocess the recordings in noisy environments.
- **Fine-tuning for Sentiment Analysis:** It fine-tunes well on subtle emotional expressions from datasets, hence improving the model for more complex sentiments such as sarcasm and ambiguity.
- **Real-time performance** can be further enhanced by utilizing GPU acceleration, along with further optimization of the inference process to make it real-time, which may reduce latency when longer audio inputs and user interaction happen at the same time.

Future Improvements

This system will realize its full potential if it undergoes the following improvements in the future:

- **Advanced Preprocessing:** Noise-cancellation techniques will be applied to increase transcription accuracy in noisy or challenging environments, thus enhancing the real-world capability of the system.

- **Extended Functionality:** By adding multi-language transcription support and sentiment analysis, this system can extend the circle of usability in different global markets and various applications.

- **Better Models:** The transformer models for emotion detection, such as Robert or other similar architecture models, would help get more depth into sentiment classification for complicated emotional contexts.

It will, therefore, be able to evolve into an even more powerful and adaptable tool based on these insights and planned enhancements, opening new ways of applying audio-based sentiment analysis to different industries.

References

1. P. Rathi, "Sentiment Analysis using BERT," Kaggle, 2021. [Online]. Available: https://www.kaggle.com/code/prakharrathi25/sentiment-analysis-using-bert. [Accessed: 21-Jan-2025].
2. "Sentiment Classification using BERT," GeeksforGeeks, 2021. [Online]. Available: https://www.geeksforgeeks.org/sentiment-classification-using-bert/. [Accessed: 21-Jan-2025].
3. M. Singh, "Sentiment Analysis with BERT using HuggingFace," Medium, 2020. [Online]. Available: https://medium.com/@manjindersingh_10145/sentiment-analysis-with-bert-using-huggingface-88e99deeec9a. [Accessed: 21-Jan-2025]
4. S. Parsh, "Audio Speech Sentiment," Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/imsparsh/audio-speech-sentiment. [Accessed: 21-Jan-2025].