Acta Scientific COMPUTER SCIENCES

Volume 4 Issue 7 July 2022

Review Article

A Call-based Natural Language Classification API Framework for Conformity Assessment of Artificial Intelligence Application in an Organization

Giuliana Gois[1], Nishant Parthasarathy[1], Ravneet Kaur Ajji[1], Shalbin Benny[1] and Parisa Naraei[2]*

[1]*Member of the Working Group, Canada*

[2]*PhD Research Investigator, Lambton College and Cestar College, Canada*

***Corresponding Author:** Parisa Naraei, PhD Research Investigator, Lambton College and Cestar College, Canada.

Received: June 02, 2022

Published: June 23, 2022

Abstract

This proposal aims to solve for the issue of customer care executives not following the compliance metrics set by the company by using Artificial intelligence to create an application that allows the companies to monitor the calls using the compliance monitor and taking the necessary steps to give better customer service.

The proposal suggests transcribing the audio files that are to be checked for compliance issues, and using Google Cloud Platform (GCP) [1] for this purpose. upon completion of this phase, the compliance monitor model will process the transcribed data and in the final phase, a manual cross referencing to be performed to ensure the absence of the compliance issue in the audio files.

The final output by the AI marks all the compliance issues along the path of the set compliance rules that was set by the organisation. Each compliance issue will be highlighted in the transcript along with a color code that will describe the compliance issue that was not followed.

Using this compliance monitor, we can assure that all calls are met with the certain standards set by the organisation and instead of checking random calls out of a pool, all calls can be monitored flawlessly

Keywords: Artificial Intelligence; Google Cloud Platform (GCP);

Introduction

The objective of the proposal is to create a compliance monitor using Artificial Intelligence to check audio files for compliance issues that are set by the organization. The goals, from a technical perspective, are broadly divided into two main categories: building the transcription function and building the compliance monitor.

The compliance monitor needs the data to be transcribed so that we can cross-reference the data with the set of compliance rules set by the organization. The major advantage of creating this system is that each organization can create its own set of compliance rules, and they can be used to check for compliance issues in their data.

The process of identifying the compliance issues can be broken down into 2 main parts: transcribing the audio parts and using natural language processing to use the compliance rules to identify the compliance issues. The transcription function converts the audio file into transcribe and then the compliance rules are monitored for the transcribed data.

Dataset

The selected dataset is the result of the browsing through various sources spread across the web for call center recordings and they have been preprocessed to focus on the audio files that worked well with the objective of the projects. The files are converted into mp3 files and stored them in Google Cloud [1].

Front-end Development

We created a frontend, figure 1, for the web application using html, bootstrap, CSS, jQuery.

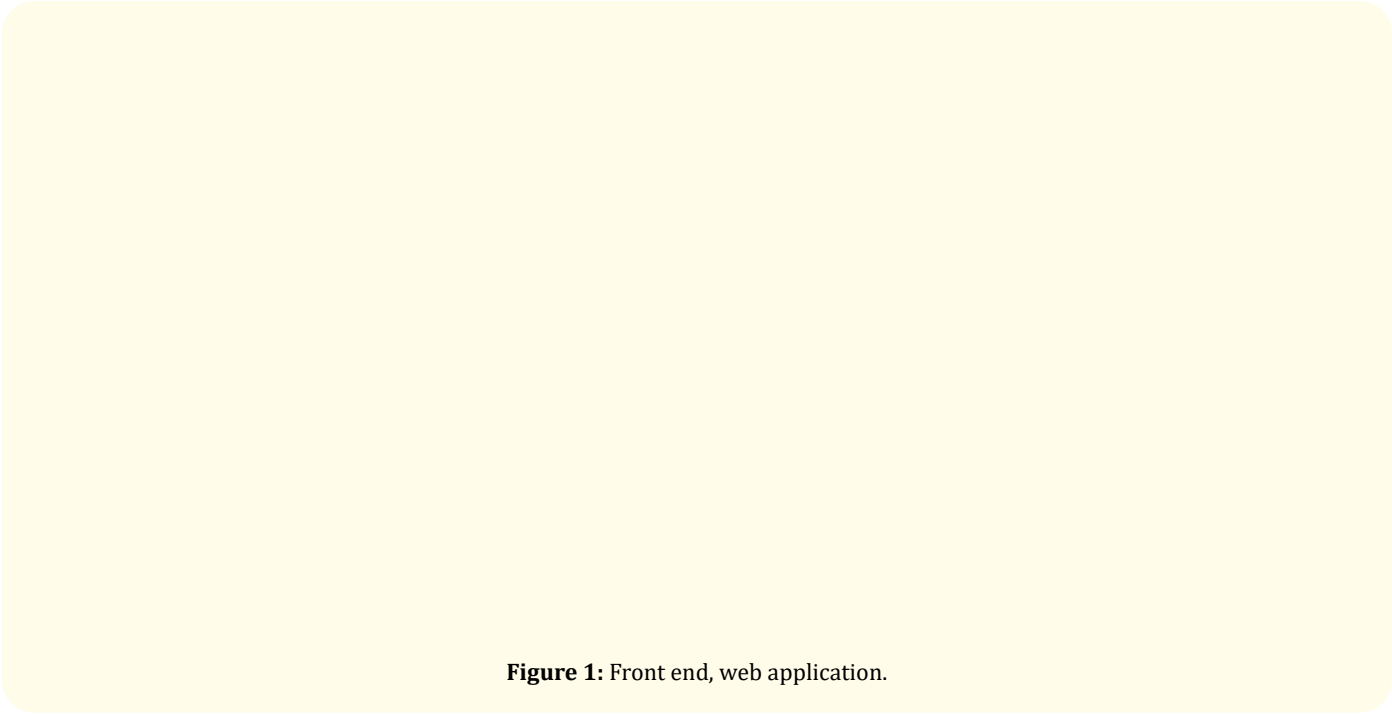

Figure 1: Front end, web application.

The input section of the webapp, asks the user to input an audio file that should be checked for compliance issues. Once the data is inserted into the application, it is passed down to the backend API, that then calls the transcribe function and the model, then once all the processes are complete, the output is displayed as two blocks of data, the first block shows the transcribed audio file and the second block shows the compliance issues.

Back-end development

The backend API was created using flask. Flask supports the API development and supports the send and receive of the data from the various models in the project.

The data flow diagram, figure 2, demonstrates the process through which the audio file is sent directly from the web interface to the transcription model. Once the transcription model performs its process, the transcribed data is stored inside the database. The stored data is then forwarded to the compliance monitor. The compliance monitor analyses the data and then the final list of compliance issues is passed back to the database where it is stored. Once the data is stored in the database, both the transcribed data and the list of compliance issues are passed down to the web application where it is displayed.

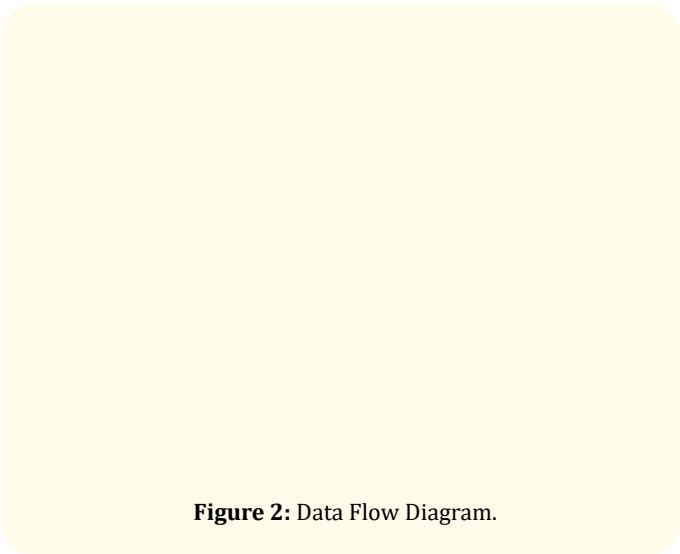

Figure 2: Data Flow Diagram.

Audio transcription

Once we receive the audio file input from the user, the API calls the transcription function. The transcription function takes the audio input from the API and stores it in the GCP bucket [1]. After the file is uploaded in GCP bucket [1], we use Google Cloud Platform (GCP), to perform the audio transcription. The files will be converted into the transcription and the transcript will be returned a text data to the API. GCP [1] is the proposed platform for the transcription due to its support for a variety of functions to explorer various aspects of the audio files and it also is an efficient library designed by google and available on demand.

Figure 3: GPU Bucket with Data.

Compliance monitor

The outline presented in figure 4, demonstrates the flow of data from audio files to checking for compliances.

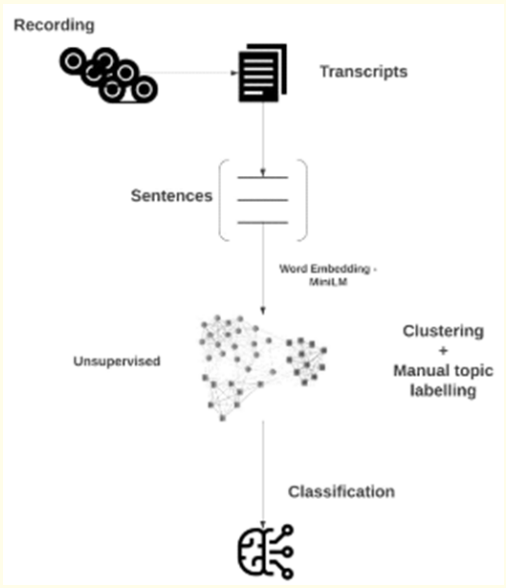

Figure 4: Compliance Monitor Outline.

There are 3 main methods in model building and the methods are listed below as follows.

Data preprocessing

Firstly, the transcribed data is sent to the model, and we use sentence tokenizer function from NLTK [2] library to convert the data into sentence, so that the data can be further processed. As the next step, the first letter of all sentences are converted into lower case and then Spacy [3] library's Natural Entity Recognition (NER) function is used to identify the entities in the data and remove all the irrelevant data from the data frame.

For word embedding, the MiniLM [4] has been selected, which is a faster method to perform word embeddings in comparison to the other well known algorithms in the field such as BERT.

Clustering

Once we receive the data frame with the word embeddings, we use FAST clustering. As a result, 322 clusters are formed, and we label the clusters manually. The resulting clusters are stored with the labels in a data frame and are passed as a JSON file to the Mongo DB which can be then accessed for model building.

Model building

We take the preprocessed data from the API to create the model. We use EDA to analyze the data frame and realize that the data is heavily imbalanced as is presented in figure 5. SMOTEENN [5] which is a sampling method is the suggested technique for performing both the oversampling and under sampling of the data. Once we resample the data, we will receive the following distribution presented in figure.

Figure 5: Data distribution after resampling.

From the pie chart, we can see that the data is distributed evenly between the four labels after the resampling of data. This would serve as the input to the proposed model.

The model follows the set of rules set by the organisation two classify into various classes of compliance issue. These compliance issues must follow the privacy guidelines set by governing bodies [10]. The compliance monitor should also consider the ethical values into consideration.

Modelling, dockerizing and deployment

A random forest model has been the selected algorithm and is the proposed algorithm to perform the classification for this task with 100% accuracy result. A docker container [6] is the proposed to serve the model on the web and Heroku [7] is the selected app deployment platform which offers a strong set of tools to ease the deployment procedure. The final product, AI compliance monitor, deployed on Heroku [7] could be accessed through this link [8]: https://ai-compliance-monitor.herokuapp.com/ and the final Project repository could be accessed through the GitHub [9].

Result and Conclusion

The proposal is the result of a proof of concept (trained, tested, dockerized and deployed model as shown in Figure 6 and Figure 7) which would enable a deeper understanding of machine learning models and how the models can be used to tackle various problems that may arise in any industry. The impact of AI in service analytics can be understood by the following result [12].

Figure 6: The final output of the AI compliance model.

Figure 7: The transcribed audio using the transcription function.

Future Work

Once the compliance monitor is deployed for any company with their own domain experts, the model can be customized according to the needs of the company. The set of rules for detecting compliance issues can be modified to implement stricter rules to implement wider array of rules and regulations. This can be further strengthened to implement best practices that can create trustworthy and robust AI system that can assure the quality and privacy of the given data [13-15]. The model can be further improved by feeding more custom data that can be used to highly increase the efficiency of the model. Furthermore, we can use this model to create live compliance checker that can act as an intermediator in between customers and executives, so that the calls can be monitored in real time and show red flags when there are any compliance inconsistencies. The compliance monitor can also be used with AI governance to implement the various governance dilemma [11].

Bibliography

1. Google. (n.d.). Google Cloud Documentation | documentation (2022).
2. NLTK. (n.d.) (2022).
3. Doc · spacy API documentation. Doc. (n.d.) (2022).
4. Wang W., *et al*. "MiniLMv2: Multi-head self-attention relation distillation for compressing pretrained transformers". arXiv.org (2021).
5. Imblearn.combine.SMOTEENN¶. imblearn.combine.SMOTEENN - imbalanced-learn 0.3.0.dev0 documentation. (n.d.). (2022).
6. Docker documentation (2022).
7. Documentation | Heroku Dev Center. (n.d.) (2022).
8. AI Compliance Monitor. (n.d.). (2022).
9. AI-Compliance-Monitor/customer-audio-complaince-check. Github. (n.d.). (2022).
10. Mökander J., *et al*. "Conformity assessments and post-market monitoring: A guide to the role of auditing in the proposed European AI Regulation". *Minds and Machines* 32.2 (2021): 241-268.
11. Mäntymäki M., *et al*. "Defining organizational AI Governance". *AI and Ethics* (2022).
12. Engel R., *et al*. "SLA-aware operational efficiency in AI-enabled service chains: Challenges ahead". *Information Systems and e-Business Management* 20.1 (2022): 199-221.

13. Pery A., *et al*. "Trustworthy artificial intelligence and process mining: Challenges and opportunities. *Lecture Notes in Business Information Processing* (2022): 395-407.

14. Shevlane T. "Structured access". *The Oxford Handbook of AI Governance* (2022).

15. Kelley S. "Employee perceptions of the effective adoption of AI Principles". *Journal of Business Ethics* (2021).