**Extracting Data from Medical Records Using NLP**

**A Major Project Report submitted in partial fulfilment of the**

**requirements for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

|  |  |
| --- | --- |
| **Submitted by** |  |
| **Taruni Sri Kadali** | **122010332048** |
| **Kadari Dinesh Reddy** | **122010307044** |
| **Leela Venkat Sai Bora** | **122010318004** |
| **Chinta Janardan Reddy** | **122010307009** |

**Under the esteemed guidance of**

**Dr. S.S Nandini**

**Assistant PROFESSOR, GIT**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GITAM**

**(Deemed to be University)**

**VISAKHAPATNAM**

**MARCH 2024**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING GITAM**

**INSTITUTE OF TECHNOLOGY**

**GITAM**

**(Deemed to be University)**

A logo with dolphins and sun

Description automatically generated

**DECLARATION**

**We, hereby declare that the Project review entitled “Extracting Data from Medical Records using NLP” is an original work done in the Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed to be University) submitted in partial fulfilment of the requirements for the award of the degree of B.Tech. in Computer Science and Engineering. The work has not been submitted to any other college or University for the award of any degree or diploma.**

|  |  |  |
| --- | --- | --- |
| **Date: 08-03-2024.** |  |  |
| **Registration No:** | **Name:** | **Signature:** |
| **122010332048** | **Taruni Sri Kadali** |  |
| **122010307044** | **Kadari Dinesh Reddy** |  |
| **122010318004** | **Leela Venkat Sai Bora** |  |
| **122010307009** | **Chinta Janardan Reddy** |  |

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GITAM INSTITUTE OF TECHNOLOGY**

**GITAM (Deemed to be University)**

A logo with dolphins and sun

Description automatically generated

**BONAFIDE CERTIFICATE**

**This is to certify that the project report entitled “Extracting Data from Medical Records using NLP” is a bonafide record of work carried out by Taruni Sri Kadali (122010332048), Kadari Dinesh Reddy (122010307044), Leela Venkat (122010318004), Chinta Janardan Reddy (122010307009) submitted in partial fulfilment of requirement for the award of degree of Bachelors of Technology in Computer Science and Engineering.**

**PROJECT GUIDE** **HEAD OF THE DEPARTMENT**

|  |  |
| --- | --- |
| **Dr.S. S NANDINI** | **Dr. SIREESHA RODDA** |
| **(ASSISTANT PROFESSOR)** | **(PROFESSOR)** |
| **CSE, GIT** | **CSE, GIT** |
| **GITAM** | **GITAM** |

**ACKNOWLEDGEMENT**

We would like to thank our project guide Dr. S S Nandini, Assistant Professor, Department of CSE for her stimulating guidance and profuse assistance. We shall always cherish our association for her guidance, encouragement, and valuable suggestions throughout the progress of this work. We consider it a great privilege to work under her guidance and constant support.

We also express our thanks to the project reviewers Dr. John Jaidhan, Professor and Dr. Raja, Professor, Department of CSE, GITAM (Deemed to be University) for their valuable suggestions and guidance for doing our project.

We consider it is a privilege to express our deepest gratitude to Dr. SIREESHA RODDA, Head of the Department, Computer Science and Engineering for her valuable suggestions and constant motivation that greatly helped us to successfully complete this project.

Finally, we deem it a great pleasure to thank one and all that helped us directly and indirectly throughout this project.

|  |  |
| --- | --- |
| Taruni Sri Kadali | 122010332048 |
| Kadari Dinesh Reddy | 122010307044 |
| Leela Venkat Sai Bora | 122010318004 |
| Chinta Janardan Reddy | 122010307009 |

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **CONTENTS** | | **PAGE.NO** |
|  | **Declaration** | | **2** |
|  | **Certificate** | | **3** |
|  | **Acknowledgment** | | **4** |
| **1.** | **Abstract** | | **6** |
| **2.** | **Introduction** | | **7** |
| **3.** | **Literature Survey** | | **8** |
| **4.** | **Problem Identification & Objectives** | | **10** |
|  | **4.1** | **– The Problem** | **10** |
|  | **4.2** | **– Proposed Method** | **10** |
| **5.** | **System Design** | | **11** |
|  | **5.1 – System Architecture** | | **11** |
|  | **5.2 – Activity Diagram** | | **14** |
|  | **5.3 – Class Diagram** | | **14** |
|  | **5.4 – Use Case Diagram** | | **15** |
|  | **5.5 – Sequence Diagram** | | **15** |
| **6.** | **Methodology** | | **16** |
|  | **6.1** | **– Data File(Kaggle)** | **20** |
|  | **6.2** | **– Results** | **21** |
| **7.** | **Conclusion** | | **23** |
| **8.** | **References** | | **24** |

1. **ABSTRACT**

Imagine trying to conduct medical research while sifting through a mountain of disorganized handwritten doctor's notes. Fortunately, Artificial Intelligence (AI) can now process this jumbled information with remarkable accuracy. Our research aimed to create an intelligent AI system that can extract meaningful health information from messy patient notes. Using 30,000 records of sample data for the analysis, we trained our AI system using NER, a tool in NLP, helps identify and group certain types of information in text to decipher symptoms, medications, and test results from these often-illegible notes. While still under development, the system demonstrated impressive potential, particularly in identifying specific health conditions.

1. **INTRODUCTION**

Extracting medical records involves using automated techniques to pinpoint and retrieve crucial patient information from their medical history. Think of it as navigating a massive pile of documents to find critical details like diagnoses, medications, allergies, and treatments. Medical record extraction streamlines this time-consuming process by converting unstructured text into organized data that can be readily examined and utilized for numerous objectives.

Medical records are written records that include personal details, health conditions, and treatment histories of people who have been in the hospital. Information is gathered from patients when they arrive at the medical facility and is kept until they leave. Anamnesis (medical history), physical exams, additional tests, illness progress, and prescribed drugs are all detailed in these files.

Medical record analysis can reveal crucial insights for diagnosing medical conditions. However, these records often contain various complex unstructured data, including abbreviations, negations, and grammatical errors. The wide variation in writing and description styles, along with potential human errors, makes data analysis challenging.

Most advancements in identifying specific entities (NER) have been made for the English language, which has more available technologies. This study aimed to create a neural network using unstructured data from medical records to extract information about symptoms, diagnoses, medications, conditions, tests, and treatments. This will make it easier to find specific details that may not be available in structured data.

1. **LITERATURE SURVEY**

In recent years, machine learning techniques have been extensively explored in Information Extraction (IE). Sequence tagging models have been widely employed, most notably Conditional Random Fields (CRFs). For named entity recognition tasks, token labeling has been the dominant approach, with CRFs performing well. However, research has demonstrated that combining BiLSTM (Bidirectional Long Short-Term Memory) with CRFs improves results. BiLSTM's ability to process sequences in both directions enhances its understanding of sequential features. By utilizing these features, CRFs can achieve more precise labeling.

Manually labeling data for training machine learning models can be challenging and time-consuming. To address this, researchers are developing techniques to automate or semi-automate the process. This is especially important in healthcare, where data types and practices are constantly changing. Automated annotation methods can help models adapt to new data more easily. Unlike rule-based approaches, machine learning models can learn complex patterns from data using automatically generated features, making them more adaptable to different data types.

In medical text analysis, research primarily focuses on identifying disease entities. Most datasets and the widely used meta-thesaurus annotate and provide information about disease entities. Extracting disease information from medical texts is crucial. Another challenge is determining named entity boundaries. Resolving this issue enhances results, especially for exact matching evaluations. However, exact matching often produces lower scores compared to partial matching. This boundary problem relates to noun phrase chunking, commonly employed in previous research.

When creating supplement resources, aiming for a resource that covers everything is not feasible. To address this, studies have explored automated updating of resources using online sources like search engines [150]. UMLS [19] and SNOMED-CT [36] are widely used additional resources that offer extensive coverage of concepts, languages, semantic types, and term information. This allows for annotating raw text and extracting more information through matching techniques. Other studies have focused on developing techniques to leverage dictionary information to compensate for limited coverage. For instance, new relationships can be derived based on synonym relationships provided by dictionaries.

Machine learning techniques can enhance dictionaries by providing features learned from them, leading to improved performance of other machine learning models [127]. The integration of machine learning and dictionaries is well-established, with machine learning enhancing the robustness of dictionaries and overcoming their limitations. These approaches explore ways to leverage the knowledge contained in ontologies. Additionally, dictionary knowledge can serve as supplemental features for machine learning models [102], providing valuable contextual information.

1. **PROBLEM IDENTIFICATION AND OBJECTIVES:**

**The Problem**

There are many ways to get medical information, such as from everyday life, the internet, and healthcare professionals. As technology in hospitals advances, there is a growing amount of medical data, much of which is in the form of text written by the author. These texts, known as clinical narratives, are often long and may contain irrelevant information. However, they are a valuable source of information for medical research and analysis, which can help inform healthcare decisions.

In healthcare, doctors require extensive knowledge of medical documentation and bear the weight of decision-making for patients. However, manually reviewing numerous documents for each patient to reach a diagnosis can be overwhelming. To address this challenge, we propose utilizing machine learning techniques and models to automate the analysis of medical records, streamlining the diagnostic process for healthcare providers.

**Proposed Method**

Ensure the consistency and quality of text and sound data by applying standardisation and preprocessing methods, eliminating any distractions or anomalies. Employ Natural Language Processing (NLP) methods, particularly Named Entity Recognition (NER), to recognise and extract important elements from medical records in text format, such as patient names, diagnoses, drugs, and other pertinent data.

**5.System Design: Proposed**

**System Architecture:**

***Data Collection:***

* Retrieves a CSV file containing medical record information, including file names, phrases, prompts, audio quality assessments, and speaker identifications.

***Data Preparation****:*

* Cleanses and prepares the data for analysis, including: - Eliminating rows with missing data.
* Extracting relevant columns from the CSV.
* Potentially creating distinct Data Frames for audio and text data, depending on their relationship and whether it warrants separation.

***Data Visualization:***

* This section uses the Seaborn library to create graphs that show: - How often certain prompts are used.
* The distribution of audio quality.

***Text Analysis (NLP):***

* This step uses the spaCy NLP library to analyze text.
* Loads a model that is trained to understand biomedical text.
* Extracts specific parts of the text and processes them using the model.
* Finds sentences and important parts of those sentences in the processed text.
* Shows a visual representation of how sentences are structured.

***Text Preprocessing:***

* This step (currently disabled) may involve obtaining a list of common words (e.g., "the") to potentially exclude from the text data.

***Audio Processing (Spectrogram Generation):*** This section aims to produce spectrograms from audio recordings:

* Functions are defined for spectrogram creation using libraries such as librosa.
* Audio files from training, validation, and test sets are processed iteratively.
* For each audio file, a mel spectrogram is generated and saved as a .jpg image (consider using a more suitable format like .png).

***Data Preparation for CNN:***This stage prepares the generated spectrograms for use in a CNN model:

* It creates an ImageDataBunch object from a Data Frame, specifying the spectrogram folder, validation split, and data transformations (potentially for normalization and resizing).
* It resizes the spectrograms to a fixed size (common for CNNs).
* It normalizes the data using ImageNet statistics (might not be ideal for spectrograms, consider custom normalization for audio data).

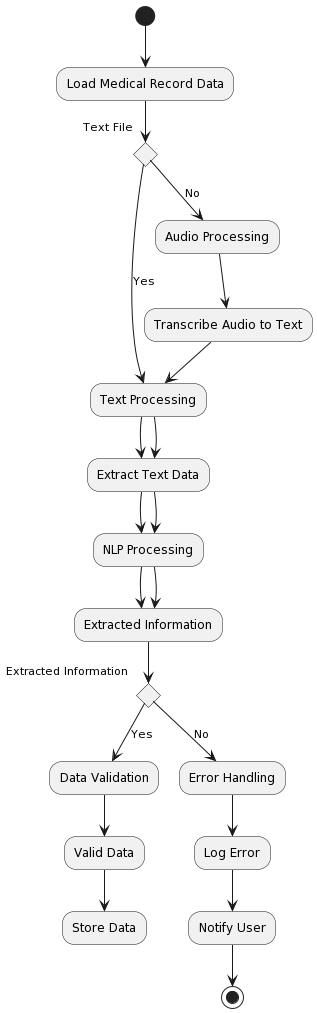
***Model Creation and Training:*** This step involves developing and training a Convolutional Neural Network (CNN) model:

* A CNN learner is constructed based on the ResNet50 architecture, utilizing previously prepared data with accuracy as the evaluation metric.
* The model undergoes training through multiple epochs, employing a one-cycle learning rate approach.
* For enhanced training, all layers are unlocked.
* An optimal learning rate is determined for fine-tuning purposes.
* The model is then subjected to additional epochs of fine-tuning.

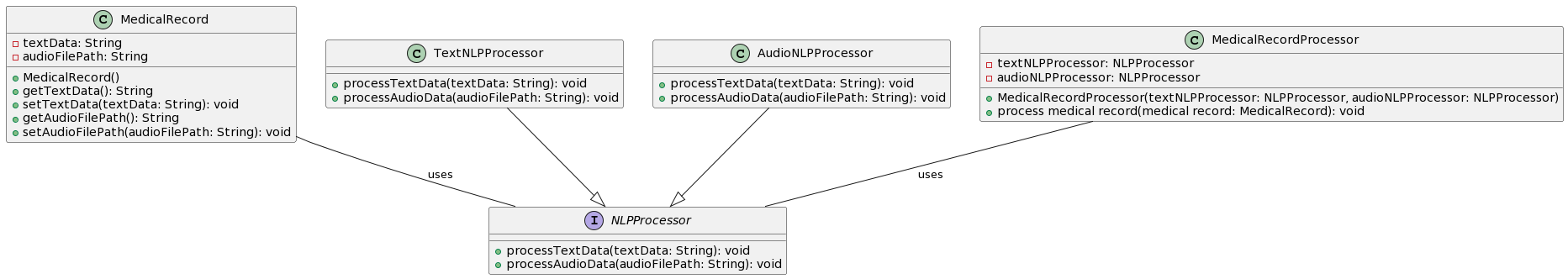
***Evaluation Phase:*** This step assesses the accuracy of the developed model:

* A "Classification Interpretation" tool examines the model's predictions.
* A confusion matrix is displayed to illustrate how effectively the model distinguishes between various audio categories.

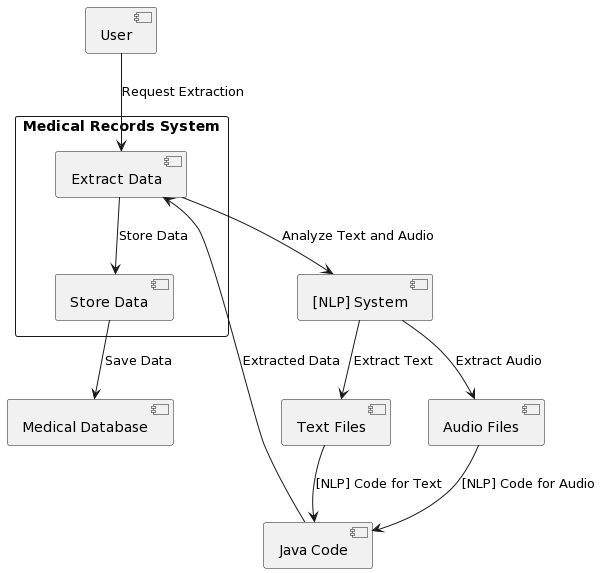
**Activity Diagram**

****

**Class Diagram**

****

**Use Case Diagram**

****

**Sequence Diagram**

**A diagram of data processing

Description automatically generated**

**6.METHODOLOGY:**

**1. Environment Setup and Library Installation:**

* !pip install scispacy, !pip install pysoundfile: These lines install the necessary libraries for medical text processing (scispacy) and audio manipulation (pysoundfile).
* !apt-get install libav-tools -y, !apt-get install zip: While these lines attempt to install libav-tools (potentially needed by other software), it's generally not required for core NLP tasks and might cause issues. Consider if other dependencies necessitate them.
* !pip freeze > '/content/drive/MyDrive/Medical Record Extraction/dockers.txt': This saves the list of installed packages to a file, potentially helpful for tracking dependencies or managing environments.
* !pip install fastai==1.0.6: You're explicitly installing FastAI version 1.0.6. Remember that TextList is not available in this version, so we'll explore alternative approaches for data loading later.
* print(fastai.\_\_version\_\_): This line verifies the installed FastAI version.

**2. Data Loading:**

* overview = pd.read\_csv(...): This reads the medical record overview CSV file containing information like filenames, phrases, prompts, audio quality ratings, and speaker IDs.
* overview = overview[['file\_name','phrase','prompt','overall\_quality\_of\_the\_audio','speaker\_id']]: This selects relevant columns from the CSV.
* overview=overview.dropna(): This removes rows with missing values.
* overviewAudio = overview[['file\_name','prompt']]: This creates a DataFrame containing filenames and prompts (potentially for audio analysis).
* overviewAudio['spec\_name'] = overviewAudio['file\_name'].str.rstrip('.wav'): This adds a column spec\_name derived from filenames, possibly for spectrogram generation.
* overviewText = overview[['phrase','prompt']]: This creates a DataFrame containing phrases and prompts (potentially for text analysis).
* There seems to be some redundancy in creating separate DataFrames for audio and text aspects. Consider combining them if the information is related.

**3. Data Cleaning and Preprocessing:**

* noNaNcsv = pd.read\_csv(...): This reads the CSV again, potentially for cleaning.
* noNaNcsv = noNaNcsv.dropna(): This removes rows with missing values (already done previously).
* noNaNcsv = noNaNcsv.to\_csv('overview-of-recordings.csv',index=False): This saves the cleaned CSV (potentially redundant if the original file is already cleaned).

**4. Data Visualization:**

* sns.set\_style("whitegrid"): This sets the style for Seaborn plots.
* promptsPlot = sns.countplot(y='prompt',data=overview): This creates a count plot showing the distribution of prompts in the data.
* qualityPlot = sns.FacetGrid(...): This creates a FacetGrid for visualizing the distribution of audio quality ratings.

**5. Text Analysis (NLP with spaCy):**

* en\_core\_sci\_sm = '/content/gdrive/MyDrive/Medical Record Extraction/en\_core\_sci\_sm-0.5.3/...': This sets the path to the pre-downloaded scispacy model for biomedical text processing.
* nlp = spacy.load(en\_core\_sci\_sm): This loads the scispacy model.
* text = overview['phrase'][62]: This selects a specific phrase from the data for analysis.
* doc = nlp(text): This processes the phrase using the spaCy model.
* print(list(doc.sents)): This prints a list of sentences identified within the phrase (might not be relevant if the phrase itself is a single sentence).
* print(doc.ents): This prints named entities (e.g., persons, locations, organizations) detected in the phrase.
* displacy.render(...): This renders the dependency parse tree of the sentence within the phrase, visually showing relationships between words. This is repeated for another example phrase (overview['phrase'][118]).

**6. Text Preprocessing and Common Word Analysis:**

* import nltk: This imports the Natural Language Toolkit (NLTK) library.
* nltk.download('stopwords'): This downloads English stopwords (common words like "the", "a", "an") for potential removal.
* plt.figure(figsize=(10,10)): This sets

**7. Creating Spectrograms:**

* testAudio = "...": This defines the path to a test audio file.
* x = create\_spectrogram(testAudio): This function (defined earlier) creates a spectrogram from the test audio and stores it in x. A spectrogram is a visual representation of the frequency and intensity of sound over time.
* The next lines load the same audio file, create a figure, generate a mel spectrogram (another type of spectrogram), and display it.
* !mkdir /kaggle/working/spectrograms: This creates a directory named spectrograms to store the generated spectrograms.

**8. Generating Training, Validation, and Test Data:**

* Data\_dir\_train, Data\_dir\_test, Data\_dir\_val: These lines use glob to create NumPy arrays containing the paths to audio files in the training, testing, and validation directories, respectively.
* The for loops iterate through these arrays:
  + Extract filename and a name for the spectrogram.
  + Call create\_melspectrogram to generate a mel spectrogram for each audio file and save it in the spectrograms directory.

**9. Data Preparation for CNN:**

* path = Path('/kaggle/working/'): Sets the path to the directory containing the generated spectrograms.
* np.random.seed(7): Sets a random seed for reproducibility.
* data = ImageDataBunch.from\_df(...): This line creates an ImageDataBunch object from a pandas DataFrame (overviewAudio).
  + df=overviewAudio: Specifies the DataFrame containing information about the audio files (potentially including prompts, which might not be directly used here).
  + folder="spectrograms": Indicates the folder containing the generated spectrograms.
  + valid\_pct=0.2: Sets the validation set to be 20% of the data.
  + suffix='.jpg': Specifies the suffix of the image files (.jpg in this case, although spectrograms are technically not images).
  + ds\_tfms=get\_transforms(): Applies data transformations (defined elsewhere, likely for normalization or resizing) to the spectrograms.
  + size=299: Resizes the spectrograms to a size of 299x299 pixels (common for CNN models like ResNet50).
  + num\_workers=0: Sets the number of worker processes for data loading (0 here).
* normalize(imagenet\_stats): Normalizes the data using ImageNet statistics (might not be ideal for spectrograms, consider custom normalization).

**10. Building and Training the CNN:**

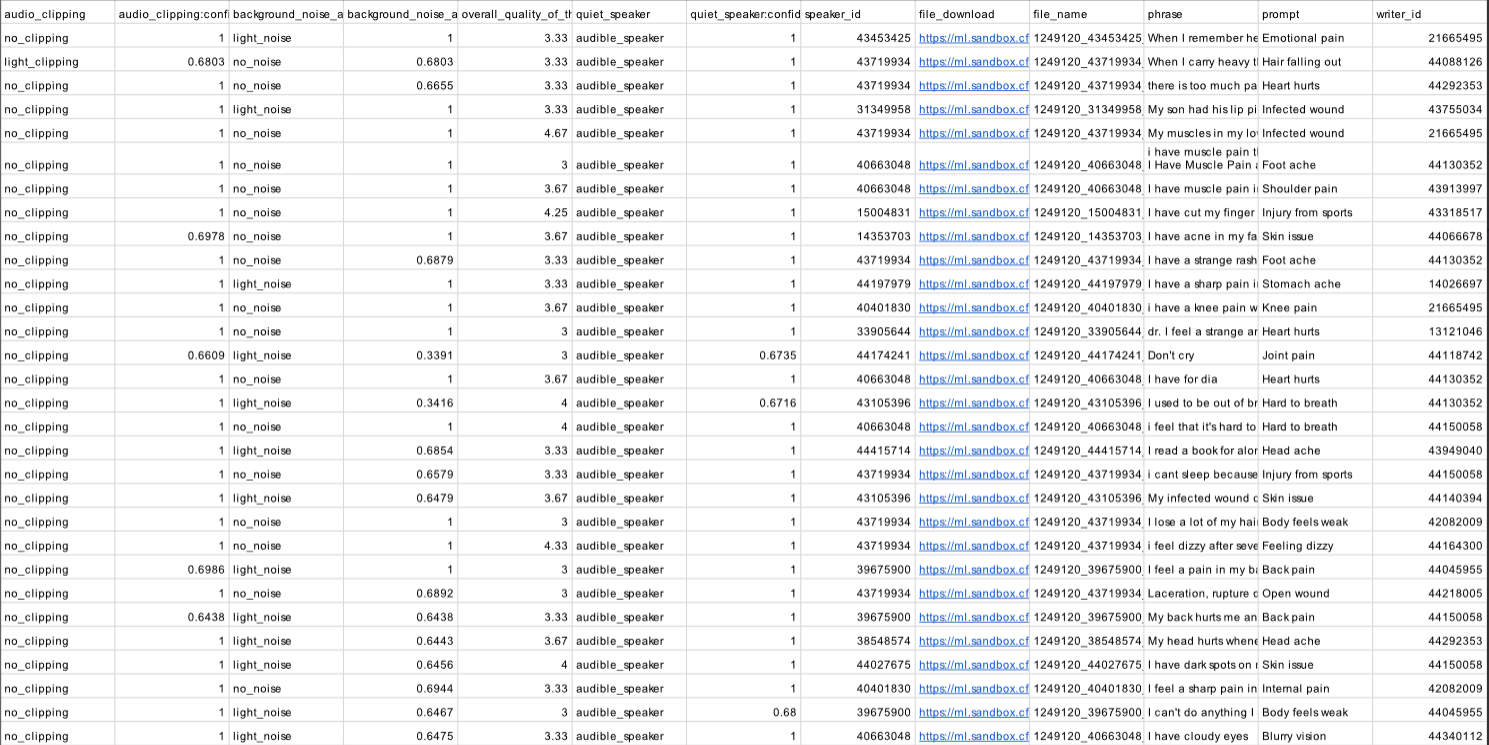
* learn = create\_cnn(data, models.resnet50, metrics=accuracy): This line creates a CNN learner using the ResNet50 architecture, the prepared data (data), and accuracy as the metric.
* learn.fit\_one\_cycle(10): Trains the model for 10 epochs using a one-cycle learning rate schedule.
* learn.unfreeze(): Unfreezes all layers of the model for further training.
* learn.lr\_find(): Finds the appropriate learning rate for fine-tuning.
  + This step might be missing output, so you might need to run it to see the learning rate plot.
* learn.recorder.plot(): Plots the training and validation loss curves.
* learn.fit\_one\_cycle(50): Fine-tunes the model for 50 epochs using the learning rate found previously.

**11. Evaluation:**

* interp = ClassificationInterpretation.from\_learner(learn): Creates a ClassificationInterpretation object from the learner.
* interp.plot\_confusion\_matrix(figsize=(10,10), dpi=60): Plots the confusion matrix to visualize the model's performance on different classes.

**Training-DataSet:**

***Text File***



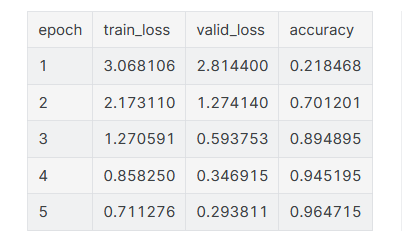
***Audio Files***

A screenshot of a computer

Description automatically generated

**RESULTS:**

Text Accuracy

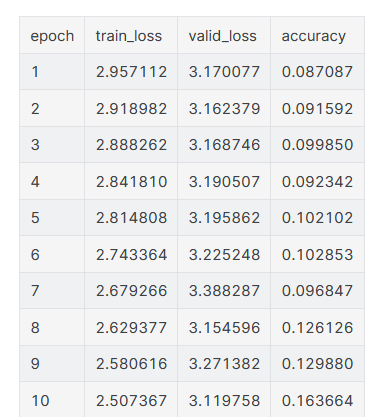


Audio Accuracy

A screenshot of a graph

Description automatically generated

Text + Audio Accuracy



1. **CONCLUSION:**

The Internet has become an invaluable resource in the medical field, enabling healthcare professionals to access vast amounts of information about diseases. This information not only enriches their knowledge but also empowers patients to seek relevant health information. For doctors, it serves as a powerful tool for making informed decisions, such as selecting the most effective treatments, prescribing appropriate medications, or identifying the underlying causes and effects of diseases. Despite the wealth of medical text data available, its sheer volume and unstructured nature make manual analysis by doctors highly challenging.

1. **REFERENCES**

* Abacha AB, Zweigenbaum P (2011) Medical entity recognition: a comparaison of semantic and statistical methods. In: Proceedings of BioNLP 2011 workshop, pp 56–64
* Aich S, Sain M, Park J, Choi KW, Kim HC (2017) A text mining approach to identify the relationship between gait-parkinson’s disease (pd) from pd based research articles. In: 2017 international conference on inventive computing and informatics (ICICI), IEEE, pp 481–485
* Akbik A, Bergmann T, Blythe D, Rasul K, Schweter S, Vollgraf R (2019) Flair: an easy-to-use framework for state-of-the-art nlp. In: Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics (Demonstrations), pp 54–59
* Al-Dafas M, Albujeer A, Hussien SA, Ibrahim RK (2022) On the adaption of data mining technology to categorize cancer diseases. Int J Artif Intell Inform 3(2):80–91
* Alex B, Grover C, Tobin R, Sudlow C, Mair G, Whiteley W (2019) Text mining brain imaging reports. J Biomed Semant 10(1):1–11
* Angeli G, Premkumar MJJ, Manning CD (2015) Leveraging linguistic structure for open domain information extraction. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing vol 1: Long Papers, pp 344–354
* Apostolova E, Channin DS, Demner-Fushman D, Furst J, Lytinen S, Raicu D (2009) Automatic segmentation of clinical texts. In: 2009 Annual international conference of the IEEE engineering in medicine and biology society, IEEE, pp 5905–5908
* Arbabi A, Adams DR, Fidler S, Brudno M (2019) Identifying clinical terms in medical text using ontology-guided machine learning. JMIR Med Inform 7(2):e12,596
* Aronson AR, Lang FM (2010) An overview of metamap: historical perspective and recent advances. J Am Med Inform Assoc 17(3):229–236
* Aydar M, Bozal O, Ozbay F (2020) Neural relation extraction: a survey. arXiv e-prints pp arXiv–2007