Tampere University

Parthasaarathy Sudarsanam

Dataset And Deep Neural Network Based Approach To Audio Question Answering

Master of Science Thesis
Faculty of Information Technology and Communication Sciences
Examiners: Prof. Tuomas Virtanen
Prof. Okko Räsänen
January 2023

ABSTRACT

Parthasaarathy Sudarsanam: Dataset and Deep Neural Network Based Approach to Audio Question Answering
Master of Science Thesis
Tampere University
Master's Degree Program in Machine Learning
January 2023

Audio question answering (AQA) is a multimodal task in which a system analyzes an audio signal and a question in natural language, to produce a desirable answer in natural language. In this thesis, a new dataset for audio question answering, Clotho-AQA, consisting of 1991 audio files each between 15 to 30 seconds in duration is presented. For each audio file in the dataset, six different questions and their corresponding answers were crowdsourced using Amazon Mechanical Turk (AMT). The questions and their corresponding answers were created by different annotators. Out of the six questions for each audio, two questions each were designed to have 'yes' and 'no' as answers respectively, while the remaining two questions have other single-word answers. For every question, answers from three independent annotators were collected. In this thesis, two baseline experiments are presented to portray the usage of the Clotho-AQA dataset - a multimodal binary classifier for 'yes' or 'no' answers and a multimodal multi-class classifier for single-word answers both based on long short-term memory (LSTM) layers. The binary classifier achieved an accuracy of 62.7% and the multi-class classifier achieved a top-1 accuracy of 54.2% and a top-5 accuracy of 93.7%. Further, an attention-based model was proposed, which increased the binary classifier accuracy to 66.2% and the top-1 and top-5 multiclass classifier accuracy to 57.5% and 99.8% respectively. Some drawbacks of the Clotho-AQA dataset such as the presence of the same answer words in different tenses, singular-plural forms, etc., that are considered as different classes for the classification problem were addressed and a refined version called Clotho-AQA_v2 is also presented. The multimodal baseline model achieved a top-1 and top-5 accuracy of 59.8% and 96.6% respectively while the attention-based model achieved a top-1 and top-5 accuracy of 61.3% and 99.6% respectively on this refined dataset. The Clotho-AQA dataset is available online here.

Keywords: Clotho-AQA, audio question answering, attention models, dataset.

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

PREFACE

I would first like to thank Prof. Tuomas Virtanen for providing me the opportunity to work as a research assistant in the Audio Research Group (ARG) at Tampere University. I would also like to thank him for being my supervisor for this thesis work and guiding me throughout various stages with his vast experience while giving me enough freedom to work independently. I express my sincere gratitude to him for being very accommodating during periods of my distress.

I also thank and acknowledge my colleagues Samuel Lipping and Dr. Konstantinos Drossos who made significant contributions during the early data collection stages of this research.

My sincere thanks to Dr. Archontis Politis, who has been my mentor for the past couple of years. I am grateful to have an approachable mentor like him around to discuss various technical and non-technical topics and answer all my queries. I would also like to thank Duygu Doğan and Shanshan Wang of the ARG for supporting me in writing my thesis. I also thank all the members of the ARG, who have been very supportive and helped me in various circumstances.

I wish to acknowledge Tampere University and CSC-IT Center for Science, Finland, for the infrastructural and computational resources used in this research.

Finally, I owe all that I am today to my family whose unconditional love and support have enabled me to pursue my goals without fear, and to my friends who have emotionally supported me throughout.

Tampere, 16th January 2023

Parthasaarathy Sudarsanam

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

AMT	Amazon mechanical turk
AQA	audio question answering
CBOW	continuous bag Of words
DAQA	diagnostic audio question answering
FFT	fast Fourier transform
GRU	gated recurrent unit
LSTM	long short-term memory
NLP	natural language processing
QA	question answering
RNN	recurrent neural network
STFT	short-time Fourier transform
VQA	visual question answering

1. INTRODUCTION

Question answering (QA) is the task of producing answers in natural language when questions are posed in natural language. Often, these questions are accompanied by a naturally occurring signal such as an image or audio and the questions posed are about the contents of these signals. If the auxiliary input is an image, the task is referred to as visual question answering (VQA) and if it is an audio signal, it is called audio question answering (AQA). Although the question answering framework is well studied for image and textual modalities, audio question answering is comparatively less explored.

Audio question answering opens the gate for new possibilities in areas such as monitoring and surveillance, machine listening, human-technology interaction, acoustical scene understanding, etc. An audio question answering system is shown in Figure 1.1. The benefit of using natural language lies in its ability to represent complex high-level information about the input signal. For example, in audio inputs, the order of events, repetition or count of events, the temporal relationship between events, etc., can be easily represented using natural language description.

The motivation for AQA comes from the widely common audio captioning task. In audio captioning, for a given input audio, the contents of the audio signal are described in natural language. There are a few drawbacks to any audio captioning system. Firstly, an audio signal can be described in multiple ways based on what the annotator chooses to focus on. Secondly, standard captioning evaluation metrics such as BLEU scores are based on n-gram equivalences. This affects the evaluation process as the system may output a new correct caption that may not be present in the ground truth or have a different sentence format. AQA is a starting point to make a system focus on a particular piece of information in audio depending on the natural language question posed to it. To

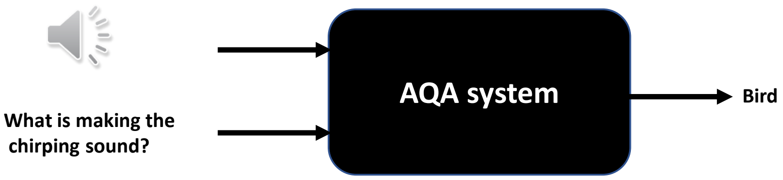


Figure 1.1. *Block diagram of an audio question answering system.*

achieve this, along with extracting interesting features from the audio, the system must also understand the textual input and find associated relationships between the audio features based on the input question to generate a natural language answer. It is hence, a very important challenge for the machine learning community in both acoustic analysis and natural language understanding.

Most QA datasets currently available are for other modalities such as textual question answering [1], [2], visual question answering [3]–[7], and video question answering [8]–[11] tasks. Most of these datasets have real-world auxiliary data. Therefore, questions and answers are annotated manually using crowdsourcing tools like Amazon Mechanical Turk (AMT). CLEAR [12] and DAQA [13] are the two popular datasets created for the AQA task. The CLEAR dataset has fixed-length audio signals of 10 different musical notes. On the other hand, DAQA contains variable-length audio signals of generic sound events. For these datasets, the audio scenes are produced synthetically by concatenating a few elementary sounds and the questions and answers are generated programmatically. While the data is generated in a constrained setup, the generated data has insufficient diversities and difficulties compared to real data. In many QA datasets, questions and answers are collected from the same annotators in the crowdsourcing framework [4], [7], [8], [10]. Some datasets such as [5], [11], use different annotators for questions and answers. Using different annotators for questions and answers ensures that only the audio signal and generic knowledge were used while answering the questions.

In this work, a new dataset called Clotho-AQA for audio question answering is introduced and machine learning models were developed to tackle this task. The dataset contains 1991 audio files chosen randomly from the Clotho dataset [14]. For each audio, six questions were collected and for each question, three answers were collected from independent annotators using AMT. Hence, each audio file is associated with 18 question-answer pairs. Developing a multimodal audio and natural language system that understands a natural language question and retrieves relevant information from the audio signal is quite challenging. Recent developments in deep learning made it an appropriate option to tackle this task. Long short-term memory (LSTM) [15] based baseline models and attention [16] based models were developed to address this task on the Clotho-AQA dataset. Please note that a part of this work including the dataset and baseline models was presented to the European conference on signal processing (EUSIPCO), 2022, with the title **Clotho-AQA: A Crowdsourced Dataset for Audio Question Answering**[17].

The remainder of this thesis is organized as follows. In Chapter 2, theoretical concepts related to signal processing and deep learning techniques used in this work are discussed. Chapter 3 focuses on related works in this field. Chapter 4 describes the data collection and data cleaning process in detail. Chapter 5 presents the baseline experiments, evaluation, and results. Finally, Chapter 6 states the conclusion of this study and possible future works.

2. THEORETICAL BACKGROUND

In this chapter, the theoretical concepts in audio signal processing, natural language processing (NLP), and deep learning used in this thesis work are described in detail. In Section 2.1 the audio signal processing concepts are discussed while in Section 2.2 and Section 2.3 natural language processing and machine learning concepts are explained respectively.

2.1 Audio Signal Processing

In this work, audio signal processing algorithms were utilized in the pre-processing stage. Specifically, mel-spectrogram audio features were extracted from input audio signals. It is essential to understand the concepts of discrete Fourier transform described in Section 2.1.1, short-time Fourier transform (STFT) explained in Section 2.1.2 and mel scale presented in Section 2.1.3 to fully understand mel-spectrogram described in Section 2.1.4.

2.1.1 Discrete Fourier Transform

A digital audio signal represented in time domain $x(n)$ describes the amplitude of the audio at each discrete time index n. All audio signals can be represented as the sum of single-frequency sine waves. The Fourier transform decomposes any given signal into its constituent single-frequency sine waves and their amplitudes i.e., it converts the time domain signal into a frequency domain signal. This frequency domain representation $X(k)$ is commonly known as the spectrum of the signal. Figure 2.1 illustrates a 10-Hz sine wave in time domain and Figure 2.2 its corresponding spectrum. The mathematical formula of the discrete Fourier transform to convert a time domain signal into its frequency domain representation is

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N}, \quad k = 0, 1, ..., N-1, \tag{2.1}$$

where N is the total number of samples in the audio signal, and j is the unit imaginary number. The fast Fourier transform (FFT) algorithm is widely used in digital signal processing to efficiently compute the discrete Fourier transform.

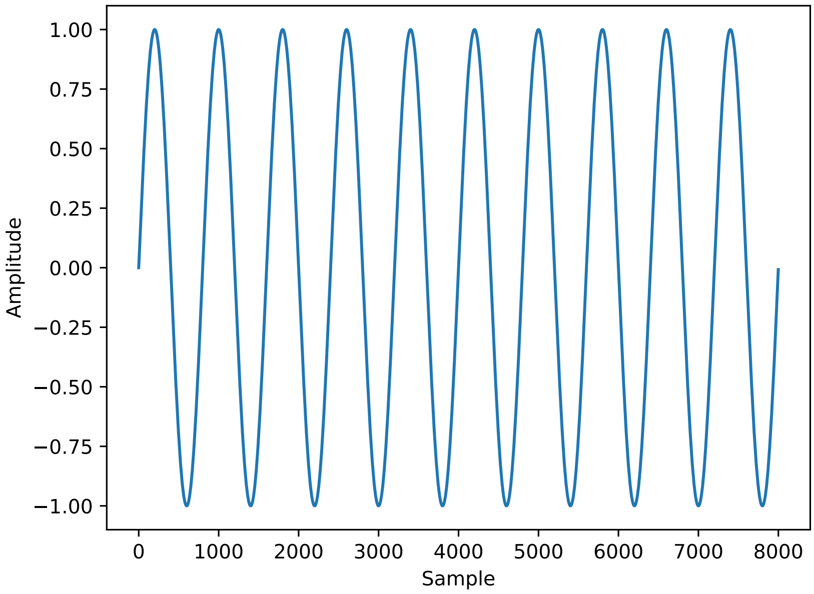


Figure 2.1. *A sine wave of frequency 10 Hz in time domain with sampling frequency 8 kHz.*

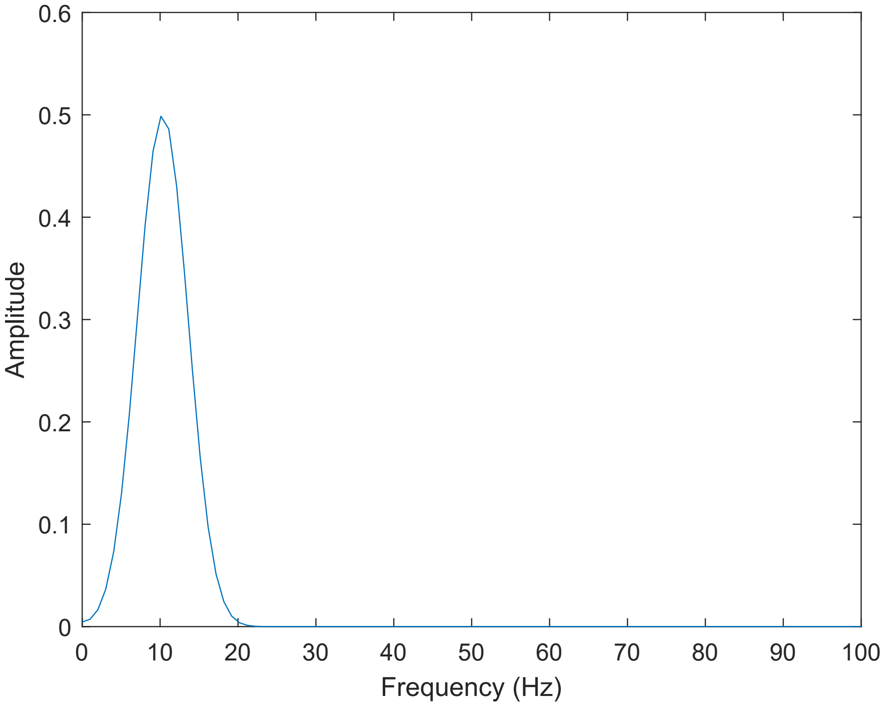


Figure 2.2. *Magnitude spectrum of the sine wave shown in Figure 2.1 having a peak at 10 Hz.*

2.1.2 Short-Time Fourier Transform

Most real-life audio signals like speech, music, etc, are non-stationary in nature. Calculating the discrete Fourier transform for the entire signal results in completely losing the time information in which different frequencies occurred in the time-domain signal. For example, consider the sine waves shown in Figure 2.3. The magnitude spectrum for both the signals as shown in Figure 2.4 is similar looking even though the frequencies clearly occur at different times in the time domain signals.

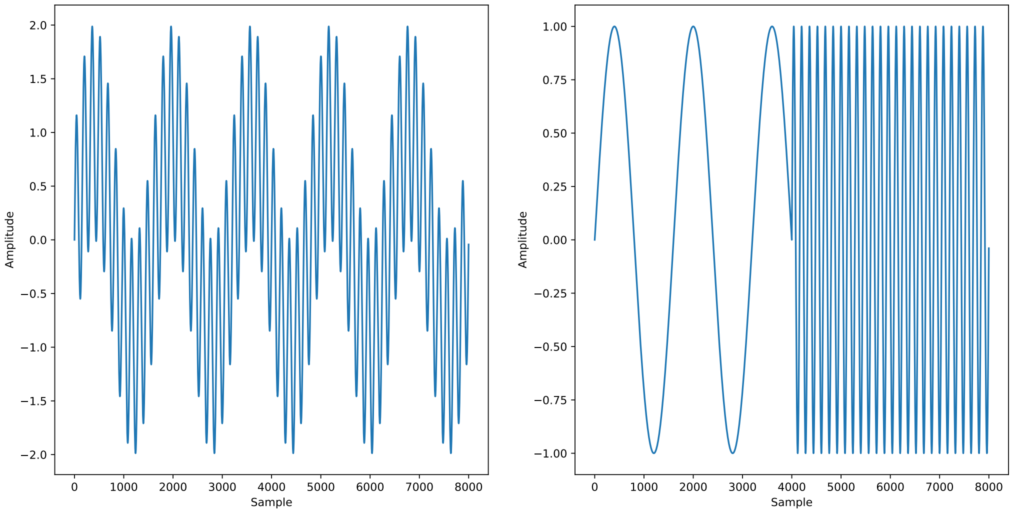


Figure 2.3. ***(Left)*** *Two sine waves with frequencies 5 Hz and 50 Hz added together.* ***(Right)*** *A sine wave with frequency 5 Hz from 0-0.5 s and 50 Hz from 0.5-1 s.*

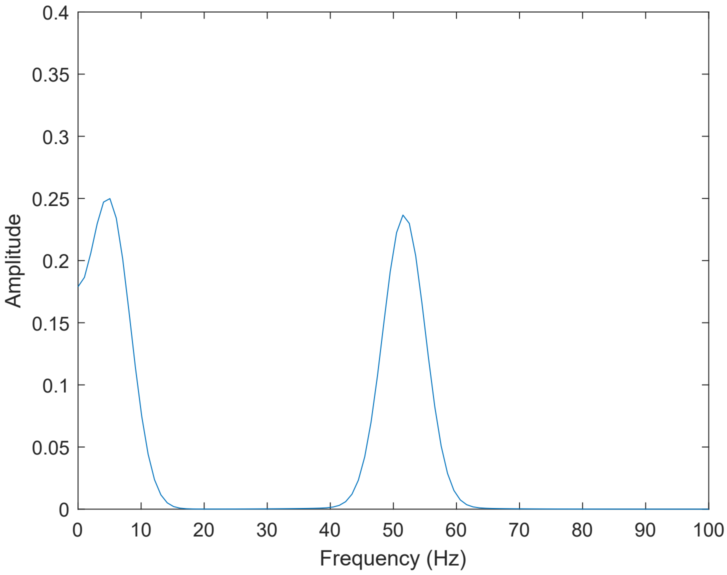


Figure 2.4. *Magnitude Spectrum of the sine waves shown in Figure 2.3 having two peaks at 5 Hz and 50 Hz.*

To overcome this problem, STFT is used. In STFT, a signal is split into shorter segments denoted by index m, each of length N. Then, FFT is applied to each of these segments. To eliminate incorrect high-frequency detections due to the abrupt splitting of segments, window filters $w(n)$ such as Hamming window or Hann window are applied to each segment before calculating the FFT. There is also overlap between each segment and the amount of overlap is determined by the hop size h. This transformation of computing the Fourier transform for overlapping windowed segments is called the short-time Fourier transform and the result $X(k, m)$ is called a spectrogram. The mathematical formula to calculate the STFT of a signal is

$$X(k, m) = \sum_{n=0}^{N-1} x(mh + n)w(n)e^{-j2\pi nk/N}, \quad k = 0, 1, ..., N-1, \tag{2.2}$$

The ability of STFT to represent both frequency and the time in which they appear in the signal is depicted clearly in Figure 2.5.

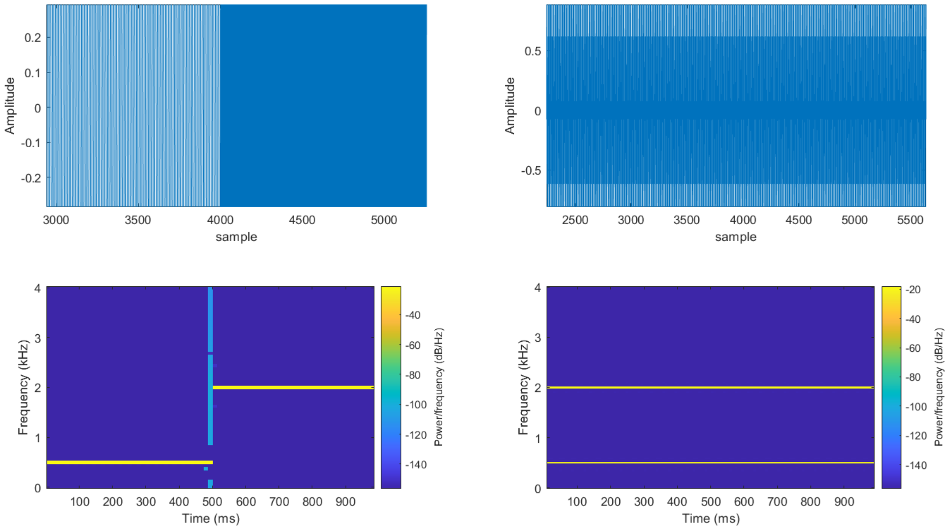


Figure 2.5. ***(Left)*** *Time domain signal and STFT of a sine wave with frequency 0.5 kHz from 0-0.5 s and 5 kHz from 0.5-1 s.* ***(Right)*** *Time domain signal and STFT of two sine waves with frequencies 0.5 kHz and 2 kHz added together.*

2.1.3 Mel Scale

It has been shown that humans do not perceive all frequencies linearly. The human ear can detect smaller changes in lower frequencies than in higher frequencies. For example, we can easily perceive the difference between 1000 Hz and 1500 Hz, but it is hard to perceive the difference between 10000 Hz and 10500 Hz, even though the frequency difference remains the same. The pitch is perceived linearly at lower frequencies and becomes logarithmic at higher frequencies. Hence, a perceptual scale of pitches such that equal distance in pitch is equally distant from one another was proposed. The mel frequency scale is shown in Figure 2.6. When the mel frequency scale is used, the frequency bands are dense at lower frequencies and spread widely at higher frequencies. Thus, a precise description of the signal is obtained with respect to human auditory perception. The conversion of frequency from the Hertz scale to the mel scale can be calculated as

$$m = 1127\, log_e(1 + f/700), \tag{2.3}$$

where f is the frequency in Hz and m is its corresponding mel frequency.

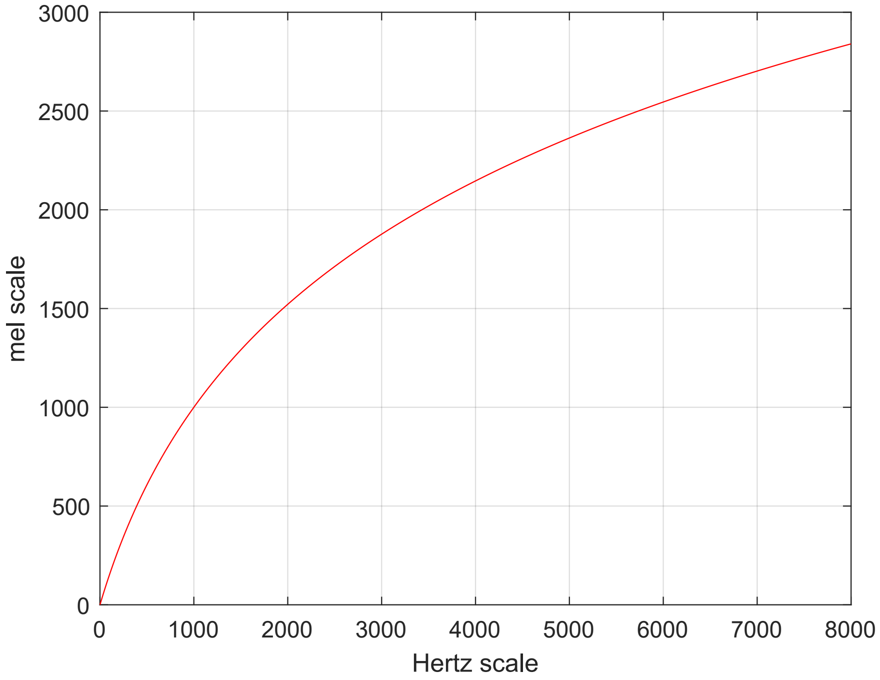


Figure 2.6. *Frequency in Hz scale along x-axis and the corresponding mel frequency along y-axis.*

2.1.4 Mel Spectrogram

The mel spectrogram of any audio signal is obtained by calculating the power spectrogram and then converting the frequencies in the resulting power spectrogram to the mel frequency scale. Firstly, mel filter bank is generated by dividing the entire frequency range into a fixed number of mel frequency bins denoted by n_{mels} (typically $n_{mels} = 64$ or 128). These frequency bins are evenly spaced based on the mel scale. The power spectrogram is obtained by calculating the absolute value of the STFT and squaring it. This power spectrogram is passed through the mel filter bank and the output from each mel filter is summed and combined to obtain the mel spectrogram of the audio. The mel spectrogram of a speech signal is shown in Figure 2.7.

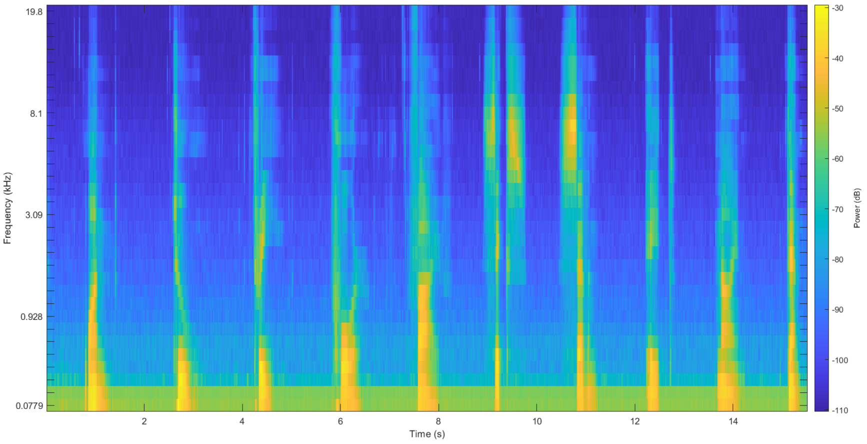


Figure 2.7. *Mel spectrogram of a sample Matlab speech audio counting from 1 to 10.*

2.2 Natural Language Processing

The question answering datasets contain questions and answers presented in natural language. It is important to represent these words as vectors so that they can be processed by machine learning models. In this section, the processes to generate these word vectors is discussed. In section 2.2.1, the concept of word vectors and one-hot vectors are described and in section 2.2.2, the continuous bag of words (CBOW) algorithm used in this work to generate the word vectors is explained in brief.

2.2.1 Word Vectors

The simplest way to represent words as vectors is to use one-hot encoding. If each word in the vocabulary is given a unique index, then in one-hot encoding, each word is represented by a unique binary vector whose values are 0 at all indices except at the index of the word.

There are two major drawbacks of this representation. Firstly, the size of each vector is equal to the size of the vocabulary. For example, there are millions of words contained in the English language. If one-hot encoding is utilized, each word is represented by million dimensional vectors. Another drawback is that all these vectors are orthogonal to each other. Hence, their similarity is always zero. These vectors do not carry any meaning about the words they encode or their semantic relationships with the other words in the vocabulary. For example, the cosine similarity between one-hot encoded word vector pairs of ‘Jungle’, ‘Lion’ and ‘Jungle’, ‘Umbrella’ are the same, zero. If the one-hot representations are

‘Jungle’ = [1, 0, 0, ..., 0]
‘Lion’ = [0, 1, 0, ..., 0]
‘Umbrella’ = [0, 0, 1, ..., 0], then
Cosine similarity (‘Jungle’, ‘Lion’) = Cosine similarity (‘Jungle’, ‘Umbrella’) = 0

2.2.2 Continuous Bag Of Words

Word2Vec is the most common technique used in NLP to generate meaningful word embeddings. It ensures that similar words or words closely related to each other have similar representations in higher dimensional space such that their cosine similarity is high. Continuous bag of words [18] and skip-gram [19] are the commonly used algorithms in Word2Vec to represent word vectors. In both these algorithms, a vector representation for each word in the vocabulary is learned as the weights of the hidden layer in a neural network.

In the CBOW task, a neural network is trained to predict a target word given its context words. The context words are the words that appear to the left and right of the target words. A’ context window’ parameter determines the number of context words to consider on each side of the target word. For example, in the sentence ‘The capital of Finland is Helsinki.’, if the target word is ‘Finland’, the context words, ‘capital’, ‘of’, ‘is, ‘Helsinki’ are fed to the model, and it is trained to predict the word ‘Finland’ in this case. This is shown in Figure 2.8. The context window in this case is two. If the target word is x_t, then two words each appearing before the target word x_{t-2}, x_{t-1} and after the target word x_{t+1}, x_{t+2} are chosen as the context. The input representation of these words are one-hot encoded vectors, and the output word is predicted using a classification task.

Once the word vectors are obtained using these word2vec algorithms, it has been found that they encode semantic relationships with other words in the vocabulary in a meaningful way. For example, the word vector representation of ‘Helsinki’ is very similar to the word vectors of other cities like ‘Stockholm’, ‘Paris’, etc, resulting in a high cosine similarity score.

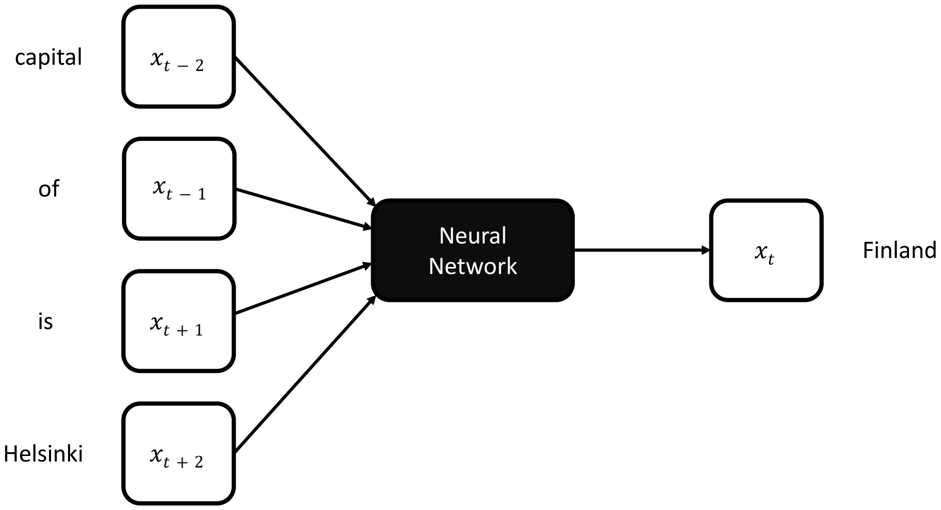


Figure 2.8. *Continuous Bag of Words algorithm to generate word vectors.*

2.3 Machine Learning Concepts

In this section, the most important machine learning and deep learning algorithms used in this thesis work are discussed. Firstly, in Section 2.3.1, the concept of supervised learning used to train neural networks is explained. A key building block of neural networks, the activation functions are described in section 2.3.2. In Section 2.3.3, the basic building block of a recurrent neural network (RNN), the vanilla RNN cell is described. In Section 2.3.4, a sophisticated component of an RNN, the LSTM cell used in this thesis work is explained briefly. Further, in Section 2.3.5, the working mechanism of an attention layer used in this work to build neural networks is discussed.

2.3.1 Supervised Learning

Machine learning systems can be trained using several algorithms such as supervised learning, semi-supervised learning, unsupervised learning, self-supervised learning, etc, depending on the properties of the dataset and the nature of the problem to be solved. In this section, the supervised learning algorithm used in this work is explained.

The supervised learning algorithm is used to train a machine learning model when the dataset contains inputs and its corresponding ground truth outputs. In a supervised learning setup, the goal of a machine learning model is to learn a mapping function that best approximates the relationship between inputs and outputs present in the dataset.

In a supervised learning setting, the dataset contains input-output pairs (x_1, y_1), (x_2, y_2), (x_3, y_3), ..., (x_n, y_n) where x_i represents the input and y_i represents its corresponding output for $i = 1, 2, ..., n$ and n is the number of data points. For example, in an image classification task, the inputs are the images and outputs represent the image classes like cat, dog, bicycle, etc. The major limitation of supervised learning is the availability of such labeled datasets on a large scale suitable for training machine learning models.

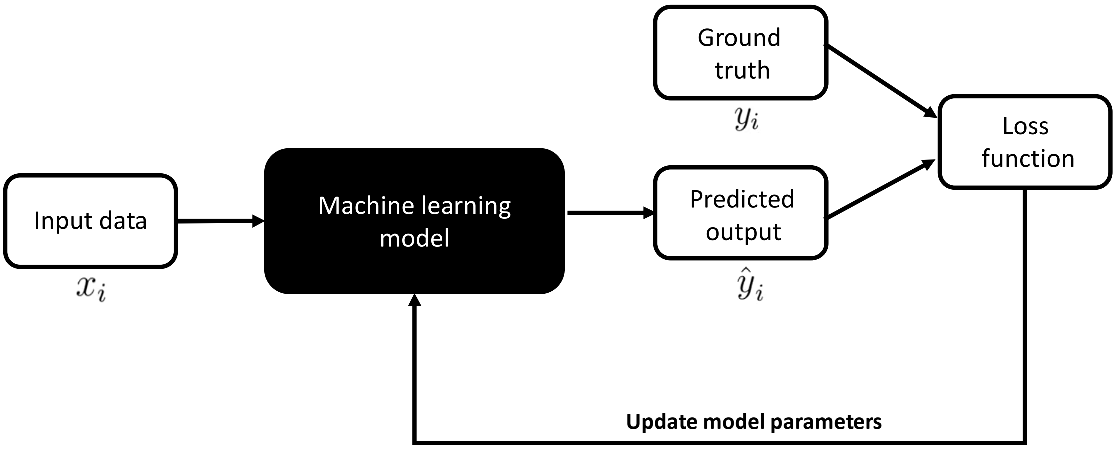


Figure 2.9. *Block diagram of a supervised learning algorithm*

Figure 2.9 shows the steps involved in supervised learning. When a neural network model is trained with a supervised learning algorithm, the parameters of a model are initialized with random weights and an input x_i is passed to the model. The input is processed by all the layers of the model and it predicts an output $\hat{y}_i$. This is known as forward propagation. Then, a loss function is used to evaluate the error between the predicted output $\hat{y}_i$ and the ground truth y_i.

The overall goal of the training process is to find suitable values for the parameters of the model such that the error between the predicted output and the ground truth is minimal. The error value calculated using the loss function is used to adjust the values of the parameters of the neural network using gradient descent algorithms. This is known as backpropagation.

These steps are repeated for all the input-output pairs and for multiple epochs until the error value is minimum. An epoch is defined as the total number of iterations in which all the training data is used once to train the machine learning model. The values of the parameters for which the error is minimum are saved and used to evaluate on unseen data which does not have any ground truth labels.

2.3.2 Activation Functions

Deep learning models generally work by extracting features from inputs and using those features to accomplish a given task. The feature extraction layers in neural networks such as convolution layers are linear operations. Hence, to stack multiple feature extraction layers to create a deep neural network, it is necessary to have non-linear operations in neural networks. Activation functions are the non-linear elements in neural networks that non-linearly scale the outputs from one layer and pass it to the next layer.

In this section, the activation functions used in the thesis work for developing neural networks are discussed. Specifically, sigmoid, rectified linear unit (ReLU), and softmax activations are explained in detail.

Sigmoid Activation

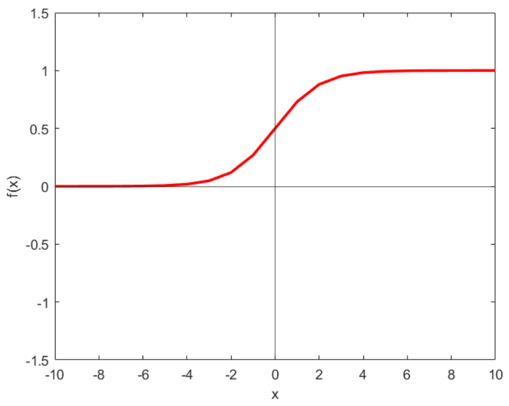


Figure 2.10. *Sigmoid activation function*

The sigmoid activation squashes the input between 0 and 1 values as shown in Figure 2.10. The sigmoid function can be expressed as

$$f(x) = \frac{1}{(1 + e^{-x})} \qquad x \in \mathbb{R} \tag{2.4}$$

The disadvantage of sigmoid activation is that it can give rise to a vanishing gradient problem. The output of the sigmoid gate saturates for large positive or negative numbers to 1 and 0, respectively. The gradient in these saturation regions is close to zero. During backpropagation, this local gradient is multiplied by the incoming gradient of this gate's output. If the local gradient is close to zero, the overall gradient becomes small and the network will not learn. This is known as the vanishing gradient problem. It is the commonly used activation function at the end of binary classifiers to produce values between 0 and 1 which can be interpreted as the probability of classes.

Rectified Linear Unit

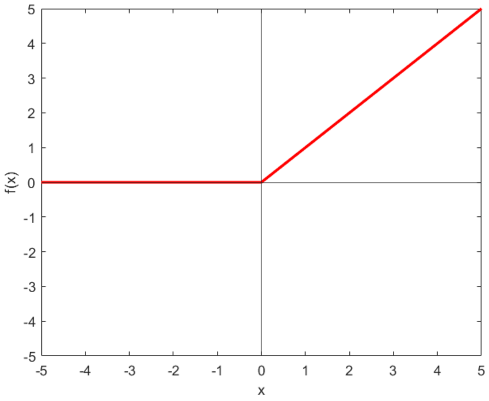


Figure 2.11. *ReLU activation function*

Rectified linear unit or ReLU shown in Figure 2.11 is the most commonly used activation function. The mathematical expression for the ReLU activation function is

$$f(x) = \begin{cases} x & x > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2.5}$$

In other words, the output is the same as the input when it is positive, otherwise, the output of ReLU is 0. The output values of ReLU are not bounded for the positive values and hence it does not have the vanishing gradient problems.

Softmax Activation

The Softmax activation function takes an input vector $[x_1, x_2, x_3, ..., x_n]$ of N real numbers and normalizes it into a probability distribution. The output vector contains N values between (0, 1) summing up to 1, proportional to the input values. Each value in the output vector computed using the softmax activation can be expressed as

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}} \qquad x_i \in \mathbb{R}. \tag{2.6}$$

It is commonly used at the end of multi-class classifiers to scale all the values of the classification layer to produce a probability distribution of the predicted classes. The index of the highest probability value after softmax is chosen as the index of the predicted class.

2.3.3 Recurrent Neural Networks

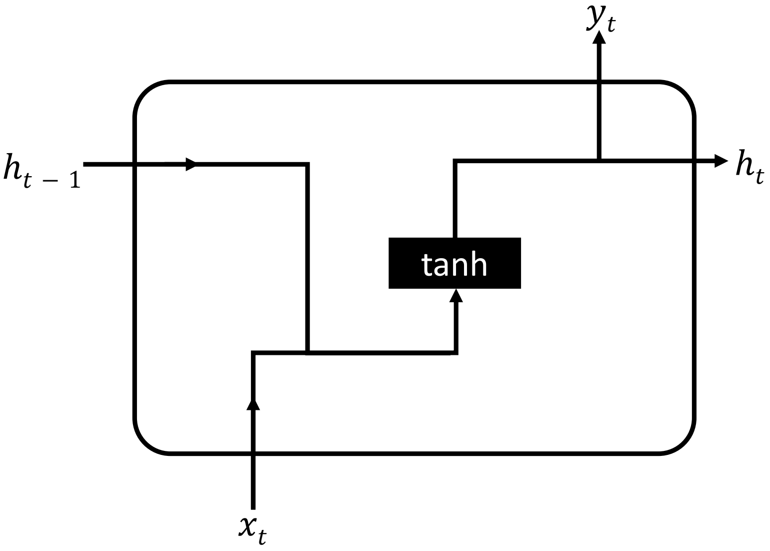


Figure 2.12. *A simple RNN cell at time step t.*

To model sequential data like text and audio, recurrent neural networks (RNNs) [20] are widely used. RNN layers process data in sequential order and they learn long-term dependencies. For example, in a task of predicting the last word of the sentence 'The capital of Finland is Helsinki', by sequentially processing the words, an RNN remembers information such as 'capital' and 'Finland' and correctly produces the word 'Helsinki' as the output. The components inside a simple RNN cell are shown in Figure 2.12. Each time step of the RNN can be represented as

$$h_t = tanh(W \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} + b), \tag{2.7}$$

where the hidden state vector from the previous time step of the RNN cell h_{t-1} is passed to the next time step and its contents are changed based on the current input word vector x_t to produce the new hidden state vector h_t. Here W and b are the weight matrix and bias vector in the RNN layer. The W and b values are learned while training the RNN using the backpropagation through time algorithm. In this algorithm, the RNN processes all the time steps sequentially and accumulates the error at each time step. After the final time step, the cumulative error for all time steps is calculated and backpropagation is performed to update the weight and bias values.

Although simple RNNs can remember contexts in smaller sentences, it is difficult to handle if the word to be predicted depends on contexts from words that appeared long before. To overcome this shortcoming, two varieties of RNN cells are commonly used, long short-term memory (LSTM) [15] and gated recurrent unit (GRU) [21]. In this work, LSTM is used for processing the sequential data.

2.3.4 Long Short-Term Memory

In vanilla RNNs, the repeating cell module has a simple $tanh$ layer. To make LSTMs better at modeling long-term dependencies, the LSTM cell has a different cell module, but it also has a repeating chain-like structure. An LSTM cell is shown in Figure 2.13.

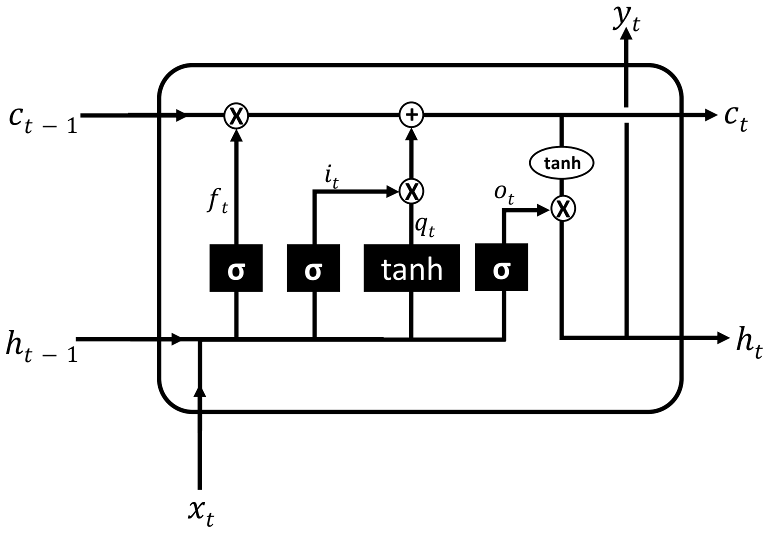


Figure 2.13. *An LSTM cell at time step t.*

The memory element in an LSTM is the cell state vector c. The cell state is passed on from one time step to another carrying information from past and present inputs. As shown in Figure, c_{t-1} is the input cell state vector to an LSTM coming from the previous time step. Inside an LSTM layer, the information present in the cell state c_{t-1} is updated using the various gates.

From left to right in Figure 2.13, the first gate is called forget gate. The functionality of the forget gate is to decide how much information present in the input cell state c_{t-1} is to be kept or removed. The sigmoid function takes the current input word vector x_t and the previous hidden state vector h_{t-1} and produces an output vector f_t, which has values between 0 and 1. This f_t is point-wise multiplied with the cell state c_{t-1}. This means if f_t has a value of 0, then all the corresponding information in the cell state vector is forgotten and if f_t is 1, all the information is retained.

The next gate is the input gate. This gate is responsible for the new information that is to be added to the cell state. It has two parts, a sigmoid function and a $tanh$ function that determines which values are to be updated and by how much. The output of this gate is also added to the cell state to produce the new cell state c_t as

$$c_t = f_t * c_{t-1} + i_t * q_t, \tag{2.8}$$

where $*$ represents the element-wise multiplication operator. Finally, each time step of the LSTM produces an output. This is controlled by the output gate. It also has two parts, a sigmoid gate to decide which parts are to be produced at the output and a tanh layer with the updated cell state c_t as input to scale the values. The output gate can be expressed as

$$h_t = o_t * tanh(c_t), \tag{2.9}$$

This output is also the hidden state passed to the next time step of the LSTM. Hence, using the different gate mechanisms, an LSTM is capable of modeling long-term dependencies compared to a vanilla RNN cell.

2.3.5 Attention Mechanism

The attention mechanism [22] allows a neural network model to attend to specific regions in the input while generating an output. Attention architectures have achieved state-of-the-art performances in various tasks ranging from natural language processing [16], to image classification [23], sound event detection [24], and sound event localization and detection [25] etc. In the case of audio question answering, it may be useful for the model to focus on specific regions of the audio signal based on the input question to produce an answer. Hence attention mechanism is very useful when designing neural network architectures for this task. Figure 2.14 shows the mechanism of an attention layer. An attention layer can be mathematically represented as

$$\mathbf{V'} = softmax(\mathbf{Q}\mathbf{K}^{\mathbf{T}})\mathbf{V}, \tag{2.10}$$

where the inputs are the three feature matrices - query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$. The query matrix is multiplied by the key matrix to determine the similarity between the query and the key. A high correspondence gives rise to a higher value and vice versa. Then the product is passed through a softmax operator explained in Section 2.3.2 to squeeze the attention scores between 0 and 1 to get an attention map W. The attention map is multiplied with the value matrix to finally produce the attention feature $\mathbf{V'}$.

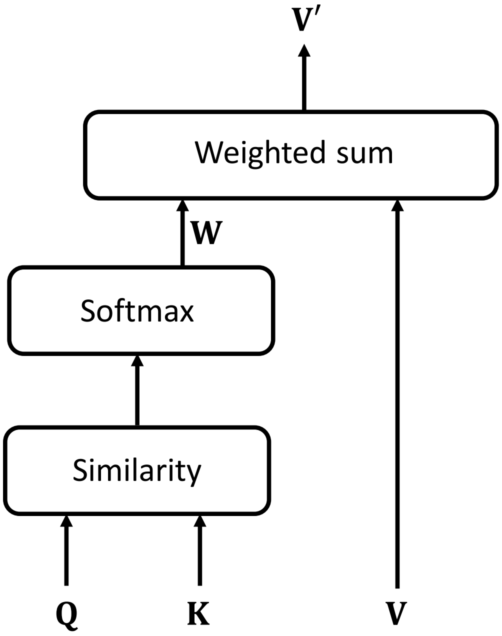


Figure 2.14. *Attention layer with inputs Query $\mathbf{Q}$, Keys $\mathbf{K}$ and Values $\mathbf{V}$ and output $\mathbf{V}'$.*

3. RELATED WORKS

In this chapter, previous literature closely related to this thesis work is briefly described. Section 3.1 and Section 3.2 review the existing CLEAR and DAQA datasets for audio question answering task, respectively. Section 3.3 introduces Clotho dataset which was used in the creation of the Clotho-AQA dataset used in this thesis work. Finally, Section 3.4 describes ideas and concepts used in the visual question answering domain which was used for the AQA task as well.

3.1 CLEAR Dataset

The first dataset introduced for the AQA task was the CLEAR [12] dataset in late 2018. The CLEAR dataset is described as a dataset for compositional language and elementary acoustic reasoning for the AQA tasks.

In the CLEAR dataset, the acoustic scenes are generated from a fixed bank of elementary sounds, similar to the CLEVR [26] data creation model used for VQA. The elementary sounds used are real recordings of musical notes from five different musical instruments namely, cello, clarinet, flute, trumpet, and violin. These recordings are obtained from the Good-Sounds database. An acoustic scene in the CLEAR dataset is created from a sequence of these elementary sounds.

The acoustic scenes of fixed length are each generated by combining 10 different elementary sounds sampled at 48 kHz. Then silent portions in the generated audio are removed by energy thresholding using a 100 ms window. To create more realistic scenes, a white uncorrelated uniform noise is added to the clean acoustic scenes.

The questions and answers are generated programmatically in the CLEAR dataset. To achieve this, each sound in an acoustic scene is indexed with the following eight attributes during the scene synthesis: **instrument family, brightness, loudness, musical note, absolute position, relative position, global position, and duration**. Questions are structured in a logical tree similar to CLEVR framework. Questions that have the same meaning are also included in the structure to increase language diversity and reduce bias. The questions are framed using the question templates where the variable part of each question is decided based on the attributes of the scene.

For example, a question related to instrument family attribute is ’What instrument plays a dark quiet sound at the end of the scene?’ and the list of unique answer words for this question includes ’cello’, ’clarinet’, ’flute’, ’trumpet’, and ’violin’. Similarly, questions are framed targeting different attributes related to the acoustic scene and there are 47 unique single-word answers in total. The CLEAR dataset was created with varying sizes such as 1000, 10000, and 50000 acoustic scenes, and 20 to 40 question answer pairs were generated for each scene.

3.2 DAQA Dataset

In 2019, the Diagnostic Audio Question Answering (DAQA) [13] dataset was released to study various aspects of temporal reasoning in the AQA framework. It contains audio sequences generated from 20 different natural sound events such as crowd applauding, dog barking, door slamming, etc. Each sound event has a source and action. For example in dog barking sounds, the source of sound is the dog and the action is barking. Out of the 20 sound events, five belong to the discrete category while the remaining are classified as continuous. If two events of the same type appear consecutively, then the listener would be able to identify them as separate events if they belong to the discrete category (for example, door slamming), while they are identified as a single event in case of continuous category (crowd applauding).

Acoustic scenes of variable length are generated by randomly concatenating 5 to 12 sound events. The length of the scenes varies from 10.5 s to 178.2 s which helps to evaluate the temporal reasoning capabilities. The audio scenes are only allowed to have consecutive occurrences of the same event if they belong to the discrete category. The sound events in an acoustic scene may also overlap by up to 500 ms. Once the clean acoustic scenes are generated, normally distributed background noise is added to 50% of the audio scenes.

Once, the audio scenes are generated, questions and answers are created programmatically. Similar to CLEAR dataset, the audio scenes are annotated with the order, identity, duration, and loudness of their individual sound events. 54 question templates are used to programmatically generate various temporal reasoning questions. Each template contains placeholders which are filled to generate the questions based on the annotations of an acoustic scene.

An example of a question template is ’What did you hear **<preposition>** the **<Source of event> <Action of event>** ?’ An example question for this template can be ’What did you hear before the dog barking?’. Each question template also has several equivalent question phrases with the same meaning to maximize language diversity. Table 3.1 shows an example of various questions and answers that are programmatically generated.

Questions	Answers
What was the shortest sound?	Door slamming
Were the first and fourth sound events the same?	No
How many times did you hear a vehicle passing by?	One
Was the first sound louder than crowd babbling?	Yes

Table 3.1. *Examples of programmatically generated questions and answers in DAQA dataset.*

There are 36 possible unique answer words in the DAQA dataset: yes, no, the 20 event types, nothing, and integers 0 to 12. Overall, the DAQA dataset contains 100000 audio clips with 80%-10%-10% splits in training, validation, and test respectively.

3.3 Clotho Dataset

The Clotho dataset [27] is an audio captioning dataset released in 2019. It contains 4981 audio files which are 15-30 s in duration. For each audio file, five captions of varying lengths between 8-20 words are collected using crowdsourcing with Amazon's mechanical turk (AMT) platform.

The audio signals along with the metadata used in the Clotho dataset were sourced from Freesound. These audio signals were selected based on the following criteria: The selected audios should have lossless file type, audio quality with a sampling frequency of at least 44.1kHz, and duration between 10 s and 300 s. The audio events were also filtered based on the tags. Audio signals with tags related to speech or music were not considered. The 10 most common tags in the Clotho dataset are ambient, water, nature, birds, noise, rain, city, wind, metal, and people. The selected audio events were then normalized to [-1, 1], the silence portion was trimmed and all the audio signals were resampled to 44.1 kHz.

The captions to the audio signals were collected by crowdsourcing using AMT. Five captions to each audio file were collected from independent annotators. Then the dataset was divided into 60% train, 20% validation, and 20% test splits based on the audio files.

Although Clotho is an audio captioning dataset, it is relevant to this thesis work, since the audio files in the newly created AQA dataset are selected from the Clotho dataset. The data splits of the AQA dataset were also created using a similar strategy used in the Clotho dataset.

3.4 Visual Question Answering

Visual question answering (VQA) is the task of providing accurate natural language answers given an input image and a natural language question about the image. There are a few popular datasets developed for the VQA task such as DAQUAR [3], Visual7W [4], COCO-QA [28], the VQA dataset **VQA**, Visual Genome [29] etc. With the exception of DAQUAR, all the VQA data sets contain images from the Microsoft Common Objects in Context (MS-COCO) [30] dataset which has 328,000 images from 91 object classes. These VQA datasets also include images from other sources such as Flickr100M data set, synthetically generated images, etc. The common approach used in these datasets to collect questions and answers is to use crowdsourcing platforms such as AMT.

The most popular among these is the VQA dataset [5]. It contains 614,163 questions and 7,984,119 answers provided by annotators for 204,721 images from the MS COCO dataset and another 150,000 questions and 1,950,000 answers for 50,000 abstract image scenes. The best-performing neural network model presented in this work was 'deeper LSTM Q + norm I' model which used a VGG net [31] to encode the images and a two-layer LSTM to encode questions. The encoded representations are fused by point-wise multiplication and then the natural language answer is obtained by passing through fully connected layers and softmax to obtain a distribution over the possible answer classes. This model produced an accuracy of 57.75% for open-ended questions and 62.70% for multiple-choice questions on the VQA dataset.

4. DATA COLLECTION

For any AQA system, the data consists of three parts. The audio signal, natural language questions, and the corresponding natural language answers. As the first step towards the task of audio question answering, a new dataset called Clotho-AQA comprising real audio signals and crowdsourced questions and answers was collected. It must be noted that this dataset was presented to the European conference on signal processing (EUSIPCO), 2022, under the title **Clotho-AQA: A Crowdsourced Dataset for Audio Question Answering [17].**

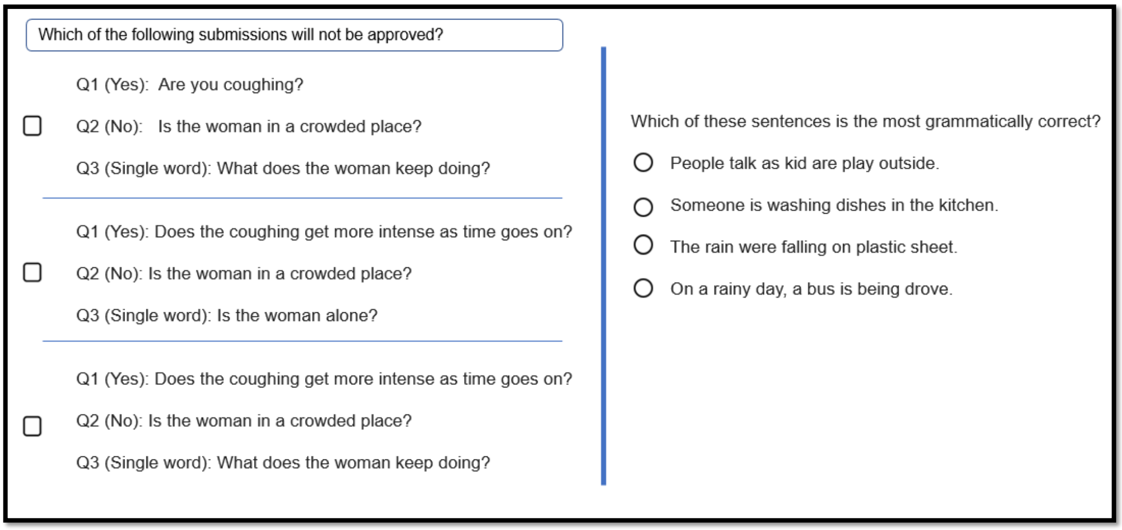
In this chapter, the data collection, cleaning process, and dataset statistics are discussed in detail. In Section 4.1 the selection of audio files is described and in section 4.2 the strategy used to crowdsource questions and answers is explained in detail. Further, in sections 4.3, 4.4 and 4.5, the techniques used for data cleaning, the algorithm for data splitting, and the dataset analysis are presented respectively.

4.1 Audio Files

As audio files for the Clotho-AQA dataset, 1991 audio files were randomly selected from the Clotho audio captioning dataset. The characteristics of the audio files present in the Clotho dataset were described in detail in section 3.3.

4.2 Questions And Answers

The questions and corresponding answers to each audio file were collected by crowdsourcing using Amazon Mechanical Turk (AMT) platform. To ensure the quality and grammatical correctness of the questions and answers, turkers with at least 3000 approved tasks and an approval rating of 95% and above were only selected. The turkers were also selected only from English-speaking areas (for example USA, UK). The question annotators were also put through a custom qualification task an example of which used to filter the turkers is shown in Figure 4.1. It checks their English grammatical skills and includes a multiple-choice question that has to be answered based on the instructions for the question annotation task shown in Figure 4.2. For example, The questions should not contain the answer word within itself (Is the car moving fast or slow?). The questions should not be addressed specifically to anyone (‘Do you like this bird chirping sound?’) and so on.



Which of the following submissions will not be approved?

□ Q1 (Yes): Are you coughing?
Q2 (No): Is the woman in a crowded place?
Q3 (Single word): What does the woman keep doing?

□ Q1 (Yes): Does the coughing get more intense as time goes on?
Q2 (No): Is the woman in a crowded place?
Q3 (Single word): Is the woman alone?

□ Q1 (Yes): Does the coughing get more intense as time goes on?
Q2 (No): Is the woman in a crowded place?
Q3 (Single word): What does the woman keep doing?

Which of these sentences is the most grammatically correct?

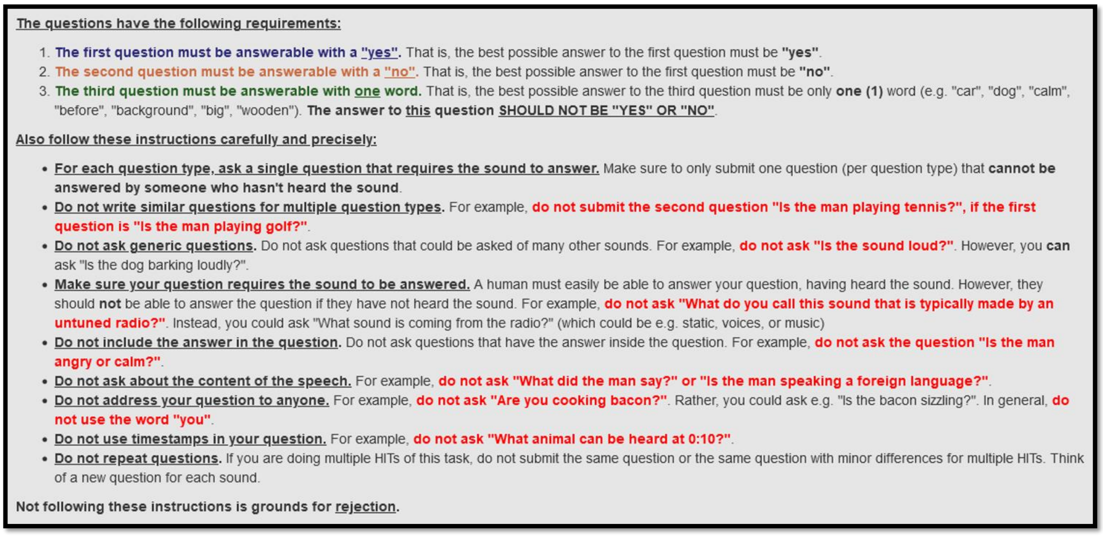
○ People talk as kid are play outside.
○ Someone is washing dishes in the kitchen.
○ The rain were falling on plastic sheet.
○ On a rainy day, a bus is being drove.

Figure 4.1. *Sample qualification test to evaluate task understanding and English grammar proficiency of turkers.*

After selecting the turkers, the questions were collected in two cycles. In the first cycle, the annotators were given access only to the audio file. No supporting information such as file names or tags associated with the audio was given. Then, three questions were collected for each audio file such that, one question should be answerable with ‘yes’, one question with ‘no’, and one question with any other single-word answer. In the second cycle, the annotators were given the audio files and the questions collected in the first cycle to avoid repetitions. Three more questions with the same ‘yes’, ‘no’ and single-word answer criteria were collected in this cycle as well. After the two question annotation cycles, there are 6 questions associated with each of the 1991 audio files.

The questions are then put through a quality check process. First, the type of questions is validated (i.e., ‘yes’ and ‘no’ questions are not single-word answer questions and vice-versa). Secondly, the contents of the questions are also checked so that they are as per the instructions given to the question annotators in Figure 4.2.

Once the questions were gathered, the next step was collecting answers to these questions. Another cycle of crowdsourcing with AMT was carried out to collect the answers. In this regard, the turkers were given the audio track and one corresponding question. For every question, answers were collected from 3 independent annotators. In addition to the answers, a confidence score was also collected from the annotators. For this, the annotators are asked “Do you feel confident that you were able to answer correctly?” and were given 3 choices ‘yes’, ‘no’, and ‘maybe’. Figure 4.3 shows an example of the instructions and answer collection window in AMT provided to the turkers for ‘yes’ or ‘no’ type questions while Figure 4.4 shows the same for single word answer type questions.



The questions have the following requirements:

1. **The first question must be answerable with a "yes".** That is, the best possible answer to the first question must be **"yes"**.
2. **The second question must be answerable with a "no".** That is, the best possible answer to the first question must be **"no"**.
3. **The third question must be answerable with one word.** That is, the best possible answer to the third question must be only **one (1)** word (e.g. "car", "dog", "calm", "before", "background", "big", "wooden"). **The answer to this question SHOULD NOT BE "YES" OR "NO"**.

Also follow these instructions carefully and precisely:

- **For each question type, ask a single question that requires the sound to answer.** Make sure to only submit one question (per question type) that **cannot be answered by someone who hasn't heard the sound**.
- **Do not write similar questions for multiple question types.** For example, **do not submit the second question "Is the man playing tennis?", if the first question is "Is the man playing golf?"**.
- **Do not ask generic questions.** Do not ask questions that could be asked of many other sounds. For example, **do not ask "Is the sound loud?"**. However, you **can** ask "Is the dog barking loudly?".
- **Make sure your question requires the sound to be answered.** A human must easily be able to answer your question, having heard the sound. However, they should **not** be able to answer the question if they have not heard the sound. For example, **do not ask "What do you call this sound that is typically made by an untuned radio?"**. Instead, you could ask "What sound is coming from the radio?" (which could be e.g. static, voices, or music)
- **Do not include the answer in the question.** Do not ask questions that have the answer inside the question. For example, **do not ask the question "Is the man angry or calm?"**.
- **Do not ask about the content of the speech.** For example, **do not ask "What did the man say?" or "Is the man speaking a foreign language?"**.
- **Do not address your question to anyone.** For example, **do not ask "Are you cooking bacon?"**. Rather, you could ask e.g. "Is the bacon sizzling?". In general, **do not use the word "you"**.
- **Do not use timestamps in your question.** For example, **do not ask "What animal can be heard at 0:10?"**.
- **Do not repeat questions.** If you are doing multiple HITs of this task, do not submit the same question or the same question with minor differences for multiple HITs. Think of a new question for each sound.

Not following these instructions is grounds for rejection.

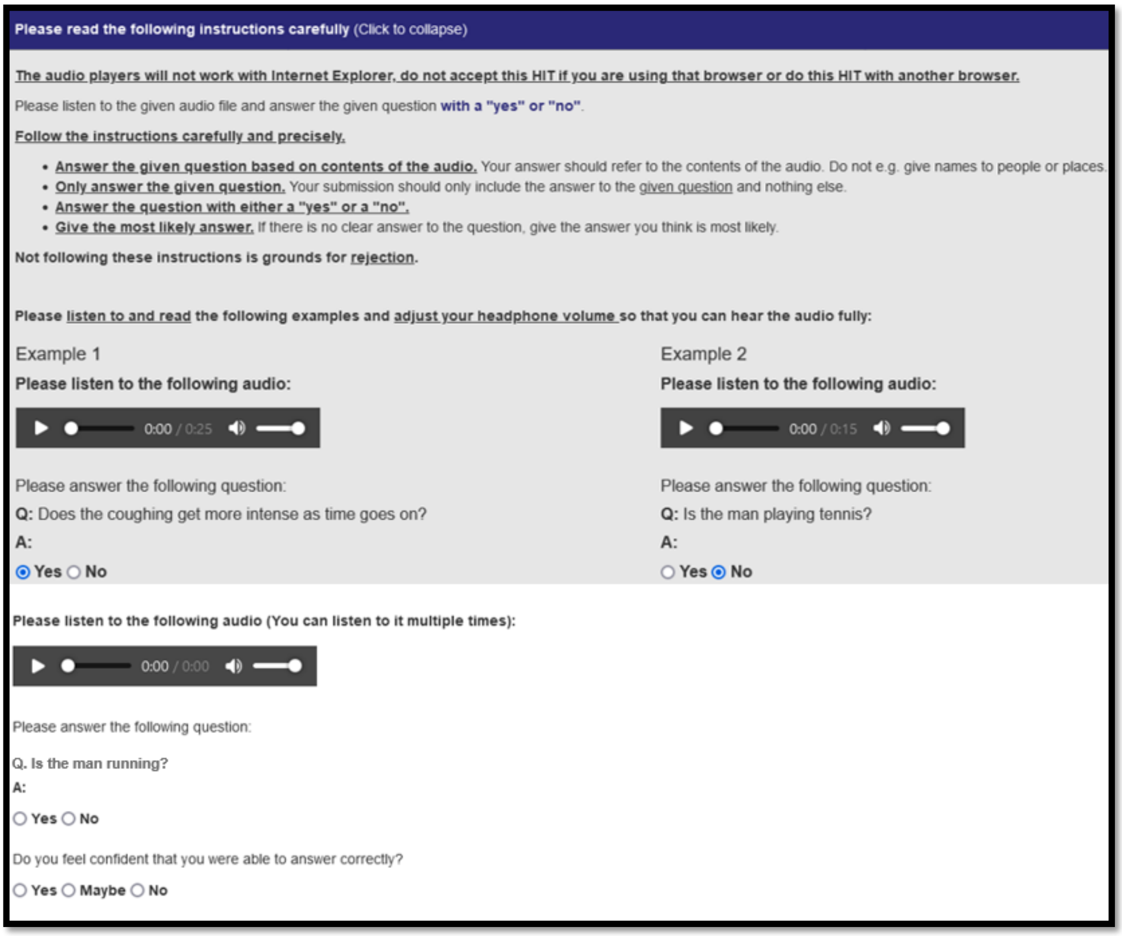
Figure 4.2. *Detailed instructions to question annotators on dos and don'ts.*

Note that the answers for 'yes' or 'no' questions were collected using radio buttons while in the case of single-word answers, the annotators can enter any single-word answers in the text box provided without any limitations. Hence there is no predefined vocabulary of answers in this dataset at the data collection stage.

After the answer collection cycle, the dataset contains 1991 audio files, 6 questions per audio file, and 3 answers each per question resulting in 18 question answer pairs for each audio file. An example of the questions and answers collected for an audio file is shown in Table 4.1.

Questions	Answers
how many birds are making noise?	two
how many birds are making noise?	two
how many birds are making noise?	several
what species of animal can be heard?	seagull
what species of animal can be heard?	bird
what species of animal can be heard?	bird
is it a dog making the noise?	no
is one bird close and one bird far away?	yes
is there a person screaming?	no
is this outside?	yes

Table 4.1. *Sample questions and answers collected for a bird chirping audio. Note that for each question answers from 3 independent annotators were collected. This table does not show all the entries.*



Please read the following instructions carefully (Click to collapse)

The audio players will not work with Internet Explorer, do not accept this HIT if you are using that browser or do this HIT with another browser.

Please listen to the given audio file and answer the given question **with a "yes" or "no"**.

Follow the instructions carefully and precisely.

- **Answer the given question based on contents of the audio.** Your answer should refer to the contents of the audio. Do not e.g. give names to people or places.
- **Only answer the given question.** Your submission should only include the answer to the given question and nothing else.
- **Answer the question with either a "yes" or a "no".**
- **Give the most likely answer.** If there is no clear answer to the question, give the answer you think is most likely.

Not following these instructions is grounds for rejection.

Please listen to and read the following examples and adjust your headphone volume so that you can hear the audio fully:

Example 1

Please listen to the following audio:

Please answer the following question:

Q: Does the coughing get more intense as time goes on?

A:

● **Yes** ○ **No**

Example 2

Please listen to the following audio:

Please answer the following question:

Q: Is the man playing tennis?

A:

○ **Yes** ● **No**

Please listen to the following audio (You can listen to it multiple times):

Please answer the following question:

Q. Is the man running?

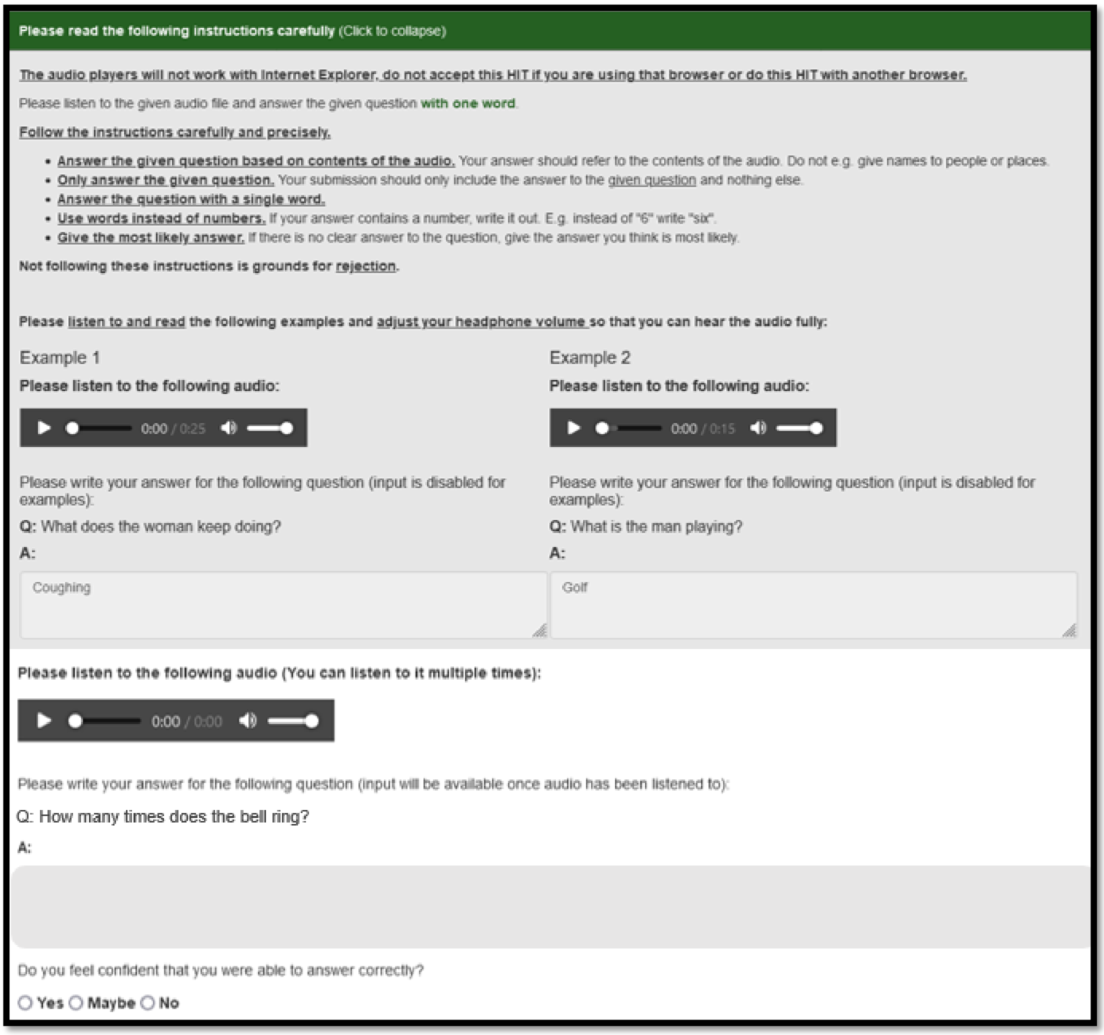
A:

○ Yes ○ No

Do you feel confident that you were able to answer correctly?

○ Yes ○ Maybe ○ No

Figure 4.3. *Detailed instructions and 'yes' or 'no' type questions answer collection window.*



Please read the following instructions carefully (Click to collapse)

The audio players will not work with Internet Explorer, do not accept this HIT if you are using that browser or do this HIT with another browser.

Please listen to the given audio file and answer the given question **with one word**.

Follow the instructions carefully and precisely.

- **Answer the given question based on contents of the audio.** Your answer should refer to the contents of the audio. Do not e.g. give names to people or places.
- **Only answer the given question.** Your submission should only include the answer to the given question and nothing else.
- **Answer the question with a single word.**
- **Use words instead of numbers.** If your answer contains a number, write it out. E.g. instead of "6" write "six".
- **Give the most likely answer.** If there is no clear answer to the question, give the answer you think is most likely.

Not following these instructions is grounds for rejection.

Please listen to and read the following examples and adjust your headphone volume so that you can hear the audio fully:

Example 1

Please listen to the following audio:

Please write your answer for the following question (input is disabled for examples):

Q: What does the woman keep doing?

A:

Coughing

Example 2

Please listen to the following audio:

Please write your answer for the following question (input is disabled for examples):

Q: What is the man playing?

A:

Golf

Please listen to the following audio (You can listen to it multiple times):

Please write your answer for the following question (input will be available once audio has been listened to):

Q: How many times does the bell ring?

A:

Do you feel confident that you were able to answer correctly?

○ **Yes** ○ **Maybe** ○ **No**

Figure 4.4. *Detailed instructions and single-word answer collection window.*

4.3 Data Cleaning

Once the answers to all questions were collected, there was a need for data cleaning so that the dataset can be used effectively to train neural networks. Two phases of data cleaning were performed. In the collected raw dataset, there were a few single-word answers that were specific to one audio file. Since the dataset is split based on the audio files, these words would appear only in one of the training, validation, or test splits. This would lead to sub-optimal training since the model will be trained with answer words that will never appear during validation or inference time or the model will never be trained with an answer word, but the answer word appears during validation or testing. To avoid this, the answer words that appear only once are replaced with other closely related answer words that are already present in the dataset. Typographical errors were also fixed in phase I of data cleaning. At the completion of this phase, the dataset only contained answers with a count of more than or equal to two. The number of unique answer words in the dataset after cleaning phase I is 830. In this thesis, this version of the dataset is referred to as Clotho-AQA_v1.

After the completion of cleaning phase I, the dataset still contained some issues relating to specificity, tense, singular and plural words, etc. Since the annotators were provided with a text box to enter their single-word answers with no restrictions on the words to use, this property crept into the dataset. Here is an example of a specificity issue from the dataset. To the question ‘Which animal is chirping?’, a few annotators simply gave ‘bird’ as their answer while a few were very specific about the species of the bird giving ‘seagull’ or ‘pigeon’ as their answers. This leads to confusion while training a neural network model for multiclass classification, as both of these answers can be considered correct. Examples of tense issues in the dataset include answer words like ‘play - playing’, ‘run - running’ etc. These words are considered completely different answers by the model. Similar issues occur with singular and plural answer words such as ‘dog – dogs’, ‘key – keys’, etc, as these are considered different from each other.

In phase II of data cleaning, these three issues were fixed. Table 4.2 shows a few examples of questions, original answers, and corrected answers. In general, the specificity issue was resolved by converting all the specific answers to their parent class. An answer class is considered specific if the dataset also contains the parent class based on the hierarchy described in wordnet. For example, ‘seagull’ was converted to ‘bird’. All the answer tenses were converted to present tense and all plural words were converted to singular. Although some of these cleaned answers do not grammatically match the question structure, it does not affect the model performance since the model considers the answer words as only classes and does not learn any language models based on them.

After cleaning phase II, the dataset contained 652 unique answer words coming down from 830 after cleaning phase I. This dataset after cleaning phase II is referred to as

Question	Original answer	Corrected answer
What species of animal can be heard?	seagull	bird
What is making the chirping sound?	parakeet	bird
What does the person do?	welding	weld
How is the weather?	rainy	rain
What instrument can be heard?	bells	bell

Table 4.2. *Examples of original and corrected answers in phase II of data cleaning*

Clotho-AQA_v2 in this thesis. Table 4.3 shows the overall dataset size after the data collection and data cleaning.

Audio files	1991
Total questions	35838
Yes/no type questions	23892
Single-word answer type questions	11946
Unique answers in Clotho-AQA_v1	830
Unique answers in Clotho-AQA_v2	652

Table 4.3. *Dataset size of Clotho-AQA after data cleaning.*

4.4 Data Splitting

The Clotho-AQA_v1 dataset is divided into non-overlapping training, validation, and testing splits based on the audio files using the ratio 60%- 20%-20%. Each unique answer in the dataset should appear in the training split and one of the validation or test splits. The Clotho dataset also used similar criteria for creating the splits. Hence, the same algorithm was followed here to split the dataset. For each audio file, a set of all unique answers associated with it is formed. Then these answer sets are used as labels for each audio file and multi-label stratification is used to create the data splits.

The final data split is performed following two steps. In the first step, 2000 splits of size 60%-40% are created. Then the top 50 splits are selected based on the split criteria, where each unique word is divided exactly 60% -40% between the two splits. In the second step, we split each of the 40% splits in step 1 into half to arrive at the final train, validation, and test splits of size 60% - 20% - 20% respectively. The split that is closest to the ideal split is taken as the final split. The number of audio files and unique answers in the dataset after splitting is shown in Table 4.4.

	Clotho-AQA_v1			Clotho-AQA_v2		
	Train	Val	Test	Train	Val	Test
Audio files	1174	344	473	1174	344	473
Unique answers	830	512	801	652	428	625

Table 4.4. *Train, validation and test splits of Clotho-AQA dataset.*

4.5 Dataset Analysis

It is important to quantitatively analyze the data, especially when used for machine learning applications. The ability of machines to learn largely depends on the quality and quantity of data used to train them. In this sub-section, various quantitative features of the dataset are described. Figure 4.5 and 4.6 shows the counts of unique answers in the Clotho-AQA_v1 and Clotho-AQA_v2 datasets plotted on a logarithmic scale respectively.

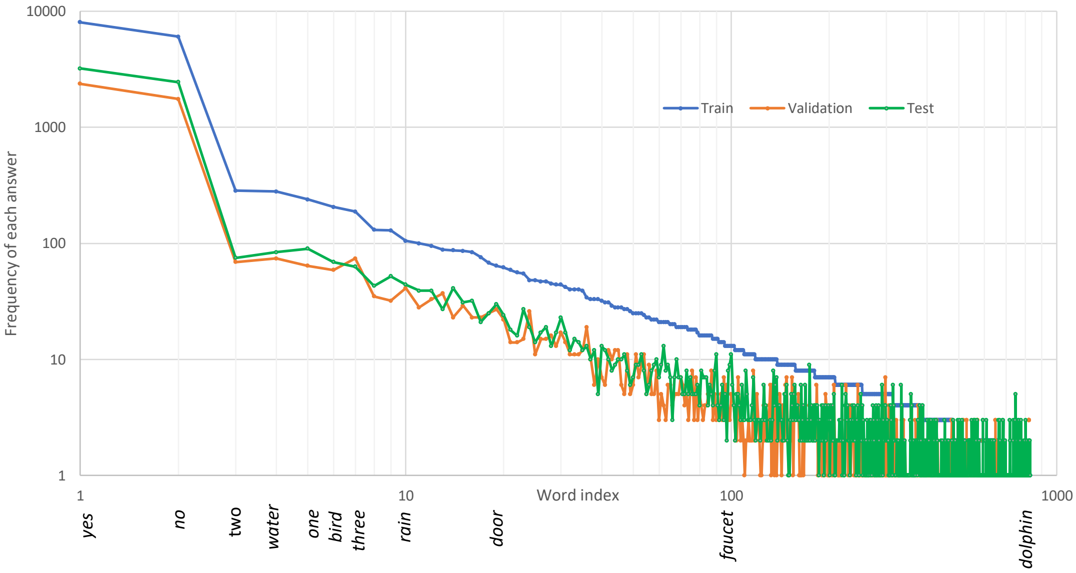


Figure 4.5. *Counts of unique answers in each of the splits of Clotho-AQA_v1. Image Source: S. Lipping, P. Sudarsanam, K. Drossos and T. Virtanen, "Clotho-AQA: A Crowdsourced Dataset for Audio Question Answering", EUSIPCO, 2022.*

As mentioned in section 4.2, for each question, answers from three independent annotators were collected. It is very useful to analyze if all three annotators gave the same answers or different answers. These statistics for all the splits of yes/no type questions in table 4.5 and for single-word answer type questions are shown in Table 4.6. Note that the yes/no questions and answers are the same in both Clotho-AQA_v1 and Clotho-AQA_v2.

In table 4.5, questions indicate the unique number of yes/no questions and **All-agree** means that all three annotators gave the same answer to a question. It is evident from the table that for approximately 59% of the yes/no type questions, all the annotators have agreed to the same answer.

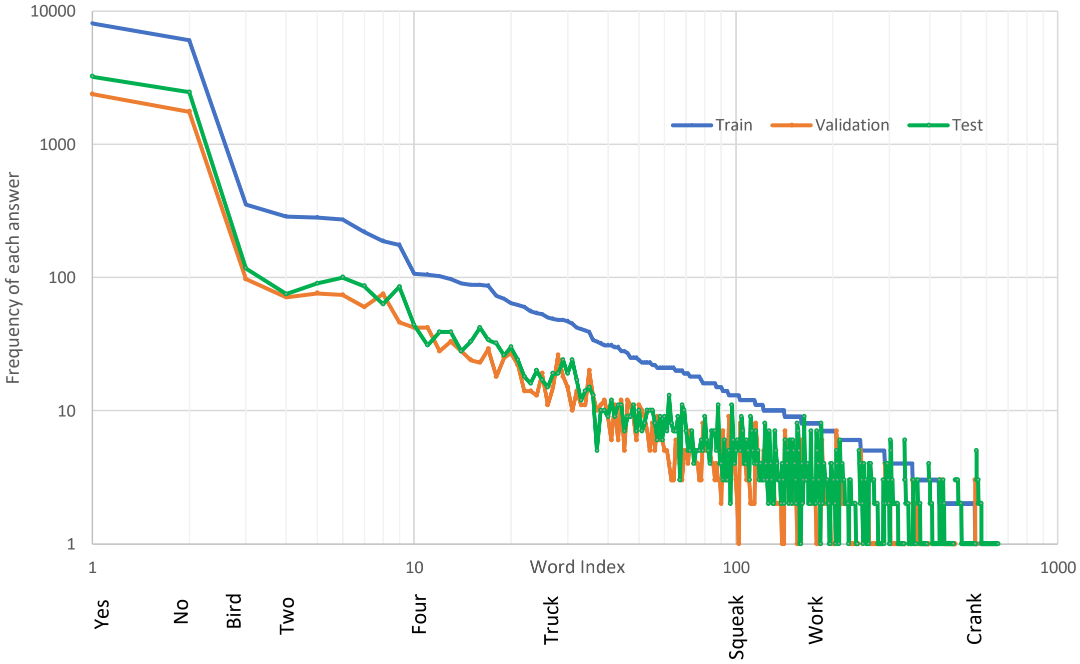


Figure 4.6. *Counts of unique answers in each of the splits of Clotho-AQA_v2*

Similarly in 4.6, questions indicate the unique number of single-word answer type questions in the dataset. **All-agree** means that the same single-word answer is given by all the answer annotators while **Majority agree** means that at least 2 out of the three annotators have given the same single-word answer. Note that the **All-agree** criterion is a subset of **Majority agree** criterion. For the remaining questions, it is understood that all the annotators gave different answers.

	Train	**Val**	**Test**
Questions	4696	1376	1892
All-agree	2766 (58.9%)	818 (59.4%)	1109 (58.6%)

Table 4.5. *Unique yes/no type questions in Clotho-AQA dataset and the number of questions for which all the annotators gave the same answer.*

	Clotho-AQA_v1			**Clotho-AQA_v2**		
	Train	**Val**	**Test**	**Train**	**Val**	**Test**
Questions	2348	688	946	2348	688	946
All-agree	509	143	203	650	184	259
Majority agree	1485	447	584	1599	480	640

Table 4.6. *Unique single-word answer type questions and a number of questions for which annotators all gave or majority gave to the same answer.*

There are also a few characteristics of the Clotho-AQA dataset which are interesting to analyze from a machine learning point of view. For example, Table 4.7 shows the number of unique answer words in the Clotho-AQA training data split that appear less frequently. This is an important factor because supervised machine learning models require more occurrences or repetitions of the answer words to learn general patterns from the data.

	Clotho-AQA_v1	Clotho-AQA_v2
Unique answers	830	652
Appears once	172	100
Appears twice	178	125
Appears thrice	99	74

Table 4.7. *Number of unique answer words with rare appearances in the training split of Clotho-AQA dataset.*

It is also interesting to analyze patterns in question phrases and their corresponding distribution of answers. Figure 4.7, 4.8, 4.9 shows the distribution of the top 10 answer words in the dataset when the question contains the word ’Weather’, ’Chirping’ and ’People’ respectively. It is interesting to note that, in cases of ’weather’ and ’Chirping’, there is an imbalance in the distribution of answers. When the question contains the word ’Weather’, close to two-thirds of the answers are ’rain’. Similarly, for ’Chirping’, more than half the answers are ’bird’. These types of distributions may pose challenges to machine learning systems since they may always output the most common answer that they find during the training process. For other words like ’people’, the top-3 answer words have a similar distribution.

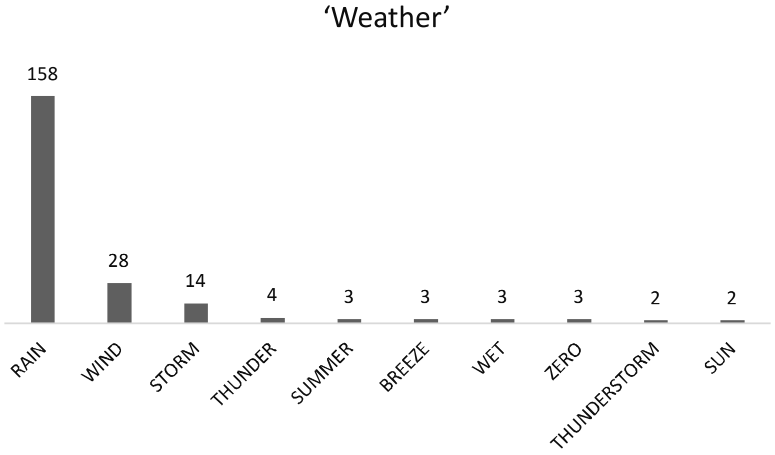


Figure 4.7. *Distribution of top-10 answer words in Clotho-AQA_v2 when the question contains ’Weather’*

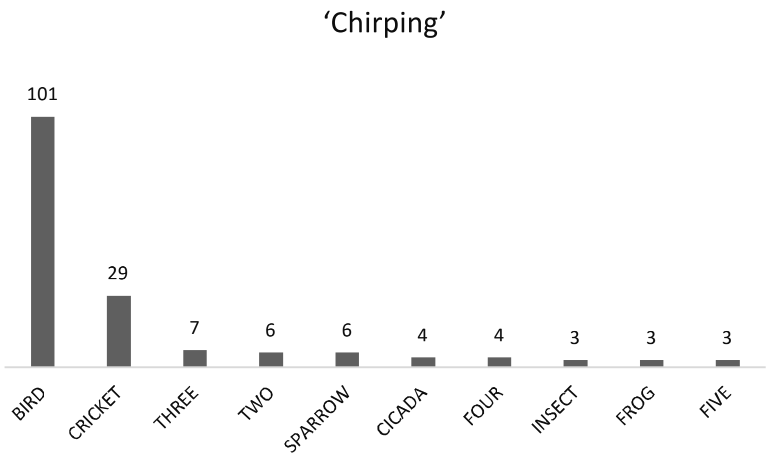


Figure 4.8. *Distribution of top-10 answer words in Clotho-AQA_v2 when the question contains 'Chirping'*

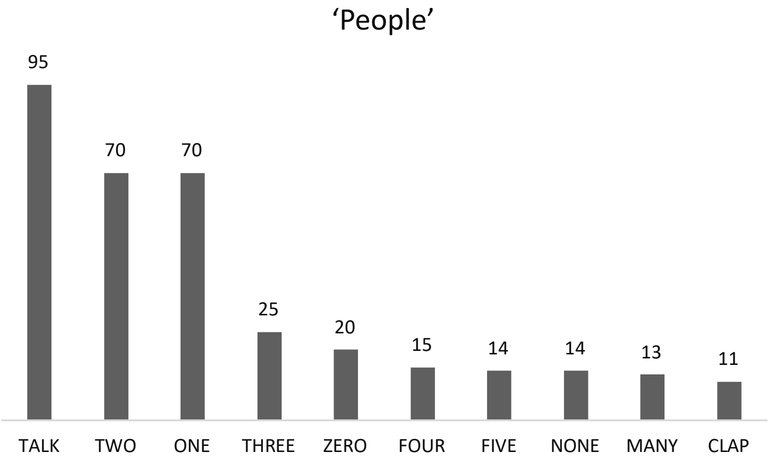


Figure 4.9. *Distribution of top-10 answer words in Clotho-AQA_v2 when the question contains 'People'*

Finally, it is also useful to know the best possible accuracy that can be achieved with the dataset. The binary test set contains 1892 unique questions. Out of these, 1109 questions have unanimous answers from all three answer annotators while the remaining 783 have different answers. Hence, an oracle model would achieve maximum accuracy of 86.2%. In the case of the Clotho-AQA_v1 test split for single-word answers, out of 946 unique questions, 203 questions have the unanimous answers provided by all the annotators and 381 questions have two out of the three annotators providing the same answer. This means that the maximum possible top-1 accuracy of the system will be 61%. Similarly, in the case of the Clotho-AQA_v2 test split for single-word answers, 259 questions have unanimous answers and 381 questions have two out of the three annotators providing the same answer leading to a maximum achievable top-1 accuracy of 65%.

5. EXPERIMENTS AND EVALUATION

In this chapter, the deep learning systems built to solve the task of audio question answering on the Clotho-AQA data described in chapter 4 are explained. As a starting point for the AQA task, the Clotho-AQA dataset only contains single-word answers. Hence, a neural network can be trained as a multi-class classifier to achieve this task. The most common approach to solve multimodal tasks is to have branches in a neural network where each branch processes an input modality to extract features and then the features are combined to accomplish the final task. The block diagram of the neural network used for this task is shown in Figure 5.1. The audio feature extractor network processes the input audio to extract relevant audio features while the textual feature extractor network does the same on the natural language question. These features are then combined and passed through a classification network that predicts a single-word answer from the list of all possible answer classes. The AQA task can be expressed as follows. If we have an audio signal A, a natural language question related to it Q, then we try to maximize the probability of producing the correct single-word answer from S, where $S = \{S1, S2, S3, \dots\}$ is a set of all unique single-word answers present in the dataset.

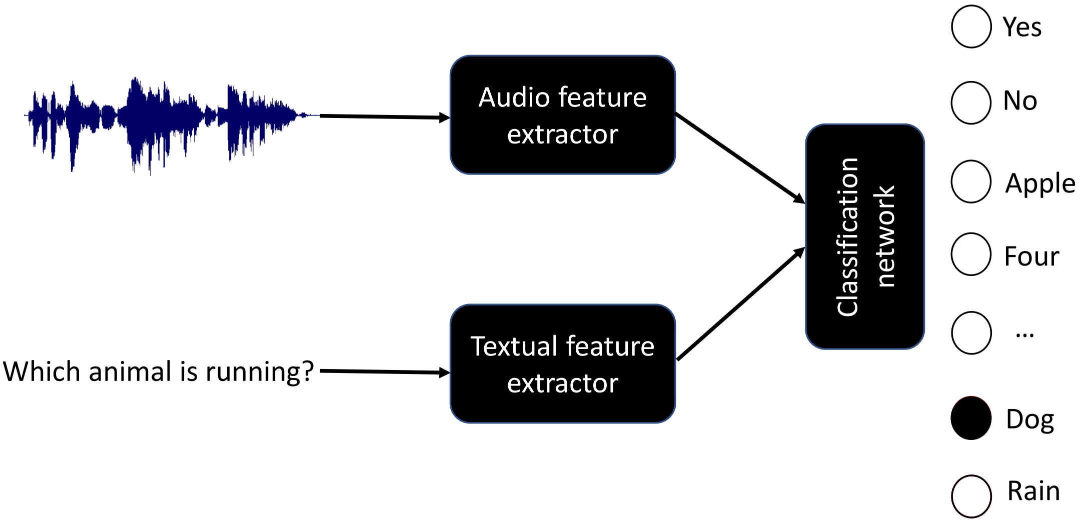


Figure 5.1. *Generic architecture of an audio question answering system.*

In section 5.1, the baseline model used to evaluate the dataset is described and in section 5.2, modern techniques in deep learning such as attention layers are added to the baseline architecture to study its impact on the AQA task. It is worth noting that some of these baseline experiments were presented to the European conference on signal processing (EUSIPCO) 2022 in the paper titled **Clotho-AQA: A Crowdsourced Dataset for Audio Question Answering [17].**

5.1 Baseline Model

Baseline models were designed to show the usability of the Clotho-AQA dataset. Since two-thirds of the dataset is dominated by ‘yes’ or ‘no’ type questions, and the remaining one-third by single-word answer type questions, we designed two baseline models to individually evaluate them. A binary classification model for the ‘yes’ or ‘no’ questions and a multi-class classification model for the single-word answers in the dataset.

The baseline model architecture used for the AQA task is shown in Figure 5.2. There are two branches in the baseline architecture, one for processing the input audio signal and the other for processing the input question posed in natural language. Hence, this model is denoted as **multimodal baseline model**. The two branches are responsible for extracting relevant features from their corresponding inputs and producing fixed-size representations of the variable-sized inputs. The features extracted from these branches are concatenated together and passed through dense layers to perform the classification task. The final classification layer contains two neurons in the case of the binary classifier and 828 and 650 classes in the case of the multi-class classifier for Clotho-AQA_v1 and Clotho-AQA_v2 respectively.

The number of audio signals and the textual questions and answers collected in the Clotho-AQA dataset are relatively small in size to learn a good representation of audio and text in itself. Hence, pre-trained models were used in the baseline to extract audio and text representations. Specifically, OpenL3 [32] was used to extract audio features while Fasttext [33] was utilized to produce pre-trained word vectors for the question text.

The OpenL3 model is based on L^3-Net [34] trained on videos from the Audioset [35] dataset. It is trained for audio-visual correspondence task in a self-supervised setting i.e, to verify if an audio input and image frames input are coming from the same video or not. The architecture of OpenL3 is shown in Figure 5.3. There were two pre-trained OpenL3 models, one trained on music videos and another trained on environmental videos. Since the audio files in the Clotho-AQA dataset contain environmental sounds, we chose the latter pre-trained model in the baseline. Since only the audio signal representation is required for the AQA task, only the audio sub-network from the OpenL3 is used.

In the baseline, the input to the OpenL3 model is $\mathbf{X} \in \mathbb{R}^{T\times128}$ mel spectrogram of the

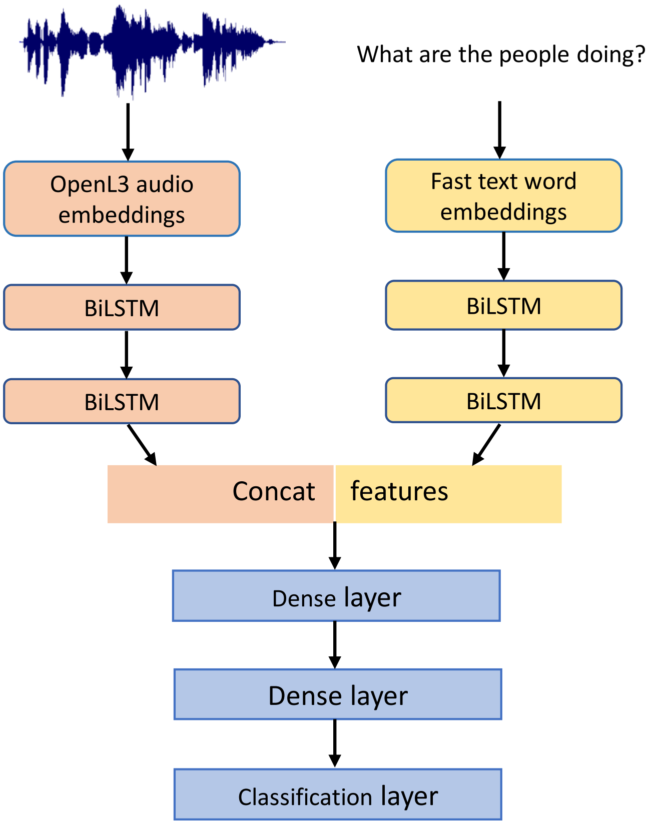


Figure 5.2. *Baseline model architecture. Image Source: S. Lipping, P. Sudarsanam, K. Drossos and T. Virtanen, "Clotho-AQA: A Crowdsourced Dataset for Audio Question Answering", EUSIPCO, 2022.*

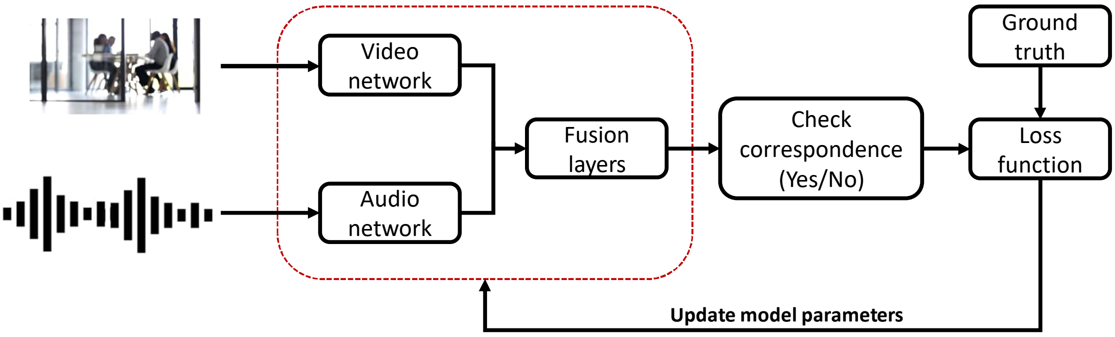


Figure 5.3. *Architecture of OpenL3 model trained for audio-visual correspondence task.*

audio with 128 mel bands and T time frames. The output from the pre-trained model is the extracted audio features $\mathbf{X}_{emb} \in \mathbb{R}^{T' \times 512}$, where T' is the number of output time frames from the OpenL3 model and 512 is the audio embedding size. These embeddings are then passed through a series of bi-directional LSTM layers BiLSTM$_n$ with $n = 1, 2,$ to learn temporal relationships and to convert the audio embeddings into a fixed size representation. The operation of the bi-directional LSTM is given by

$$\mathbf{X}_n = \text{BiLSTM}_n(\mathbf{X}_{n-1}), \tag{5.1}$$

where $\mathbf{X}_0$ = $\mathbf{X}_{emb}$. If h is the number of hidden units in the Bi-LSTM, then $\mathbf{X}_n \in \mathbb{R}^{T' \times 2h}$. The final time step of the last BiLSTM layer $\mathbf{x}_n \in \mathbb{R}^{2h}$, is chosen as the output to represent the fixed size audio embedding.

Similarly, for the textual question input, the input words are converted into word vectors using the pre-trained word vectors from Fasttext. The Fasttext word vectors were computed by training a CBOW Word2Vec model as explained in section 2.2.2 on Wikipedia 2017, UMBC webbase corpus, and statmt.org datasets.

The input to the Fasttext model is the natural language question $\mathbf{Q}$. If the question $\mathbf{Q}$ has K words, then the output word-embeddings for the question is $\mathbf{Q}_{emb} \in \mathbb{R}^{K \times 300}$, where 300 is the size of the word embedding. These output embeddings are also passed through a series of bi-directional LSTM layers to generate fixed-size representations of the questions independent of the number of words in the question. If h' is the size of the hidden units in the BiLSTM layer, the final time step of the last BiLSTM layer $\mathbf{q}_n \in \mathbb{R}^{2h'}$, is considered to represent the question embedding.

In the case of the binary classifier, the hidden size of the bidirectional LSTM layers was fixed to 128 with a dropout of 0.2 for both the audio and text branches. In the case of the multiclass classifier, the hidden size was 512 for both branches. These parameters were tuned based on the performance of the baseline model on the validation split.

The audio and question representations from both the branches were then concatenated and processed by a series of dense neural network layers Dense$_k$ with $k = 1, 2$ with ReLU non-linearity. A dense layer also called a fully connected layer is a layer in which every neuron is connected to every other neuron from the previous layer. These dense layers combine the learned features from both branches which are useful for the classification task. The dense layers are given by

$$\mathbf{D}_k = \text{Dense}_k(\mathbf{D}_{k-1}), \tag{5.2}$$

where $\mathbf{D}_0 = [\mathbf{x}_n; \mathbf{q}_n]$. There are 256 and 128 neurons respectively in the two dense layers in the binary classifier, while there are 1024 neurons each, in the case of the multi-class classifier. The increased size of the dense layers in multi-class classifiers improves the model's capacity to capture finer details to classify among a large number of different answer classes.

The output from the final dense layer is fed to the classification layer, which is another dense layer with neurons equal to the number of unique answer words. The output of this classification layer is $\hat{y} \in \mathbb{R}^C$, where C is the number of answer classes. The classification layer can be considered as a simple logistic regressor in case of the binary 'yes' or 'no' classification, while it contains 828 and 650 neurons in case of multi-class classification on Clotho-AQA_v1 and Clotho-AQA_v2 respectively. The classification layer is followed by a sigmoid activation function in the binary classifier and a softmax activation in the multi-class classifier to calculate the probabilities of each answer.

5.2 Attention Based Model

Cross-attention layers were introduced in the baseline model architecture to study its impact on the AQA task. In the baseline architecture, it can be seen that the audio and textual features are computed independently and they are simply concatenated and fed to the classifier. The general idea behind the introduction of cross-attention layers is that they can look for specific features in the audio representation that are closely related to the natural language words in the question. Thus the joint representation is taken from the output of the attention layer instead of the simple concatenation used in the baseline. Hence, this model is denoted as **multimodal cross attention model**. The architecture of the attention based model is shown in Figure 5.4.

In the attention based model, the output of the audio Bi-LSTM units is $\mathbf{X}_n \in \mathbb{R}^{T' \times 2h}$, where h is the number of hidden units in the Bi-LSTM layer and T' is the number of output frames from the OpenL3 model as shown in Equation 2.3.4. Similarly, the output of the textual branch Bi-LSTM is $\mathbf{Q}_n \in \mathbb{R}^{K \times 2h'}$, where h' is the number of hidden units in the Bi-LSTM layer and K is the number of words in the natural language question. For each of these K vectors, attention is applied on the audio features to determine which audio features are important to each of the question words to decide the answer. Hence, the output of the textual Bi-LSTM layers is used as the **query** input while the output from the audio Bi-LSTM layers is used as **key** and **value** inputs to the attention layer as explained in section 2.3.5. Two layers of cross attention are used with residual connections as shown in Figure 5.4. Residual connections connect outputs from an earlier layer to a future layer skipping one or more layers in between to provide an alternate path. In this architecture, the output of the final text Bi-LSTM layer is added to the output of the multi-head attention layer using a residual connection. The output of the cross attention layer is $\mathbf{V}' \in \mathbb{R}^{K \times m}$, where m is the attention size. In all our experiments, the attention size is fixed at 1024 with 8 attention heads. In an attention layer with multiple heads,

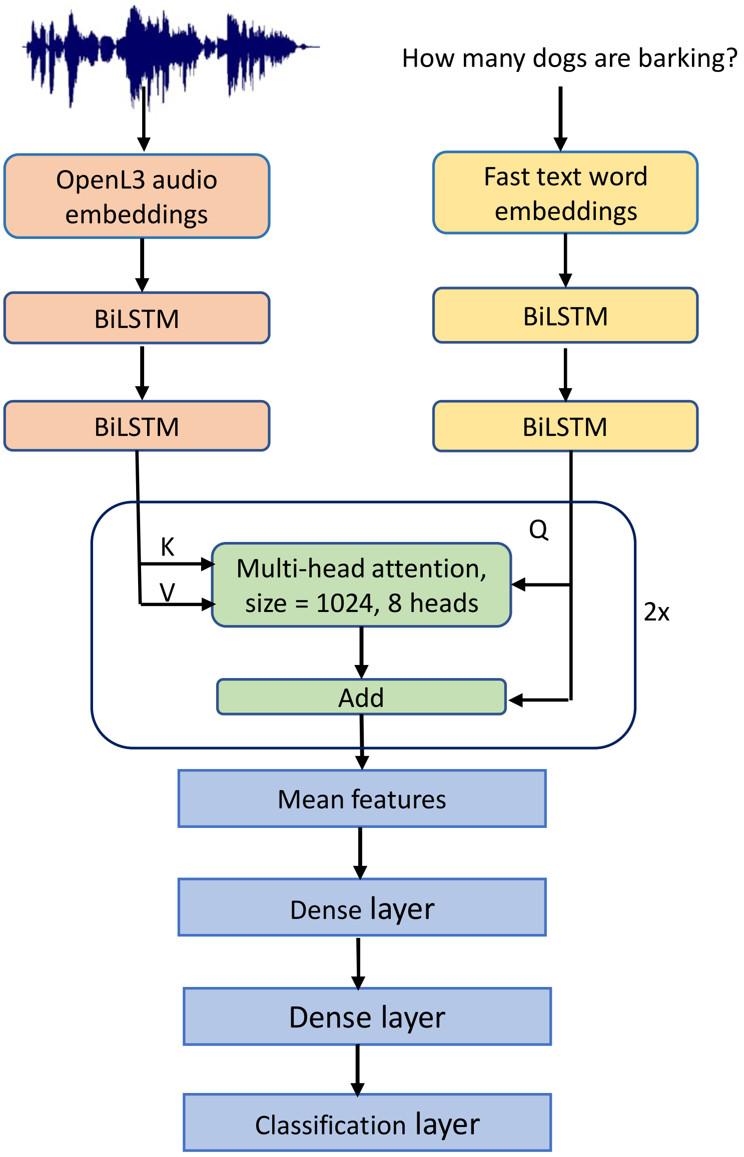


Figure 5.4. *Attention model architecture*

each head computes the attention independently in parallel and all the attention outputs are concatenated and passed through a linear layer to produce the final output of the attention layer. This attention layer is also known as multi-head attention layer. To obtain a fixed size representation, the mean is taken over the words axis of the output of the multi-head attention layer to produce $\mathbf{v}' \in \mathbb{R}^m$. This is then passed through dense layers for classification similar to the baseline architecture.

5.3 Evaluation

The baseline and the attention based models were trained and evaluated on both the Clotho-AQA_v1 and Clotho-AQA_v2 data splits obtained as mentioned in section 4.4. The datasets contain 18 question and answer pairs for each audio file. The data splits for the binary classifier are obtained by selecting the ‘yes’ or ‘no’ questions from the respective data splits. As a result, the number of audio files is 1174, 344, and 473 for training, validation, and test splits respectively. Each audio file is associated with 12 ‘yes’ or ‘no’ question-answer pairs. Similarly, the dataset is obtained for the multi-class classification by selecting the single-word answers from the respective data splits. This resulted in the same number of audio files as the original splits with each having six question and answer pairs.

The performance of the binary classifier is analyzed on contradicting answers provided by different annotators to the same question. In this regard, three cases are considered. In the first case, all the question-answer pairs are considered valid if they contain contradicting answers. In the second case, only those question-answer pairs for which all three annotators have responded unanimously are considered valid. In the third case, a majority voting scheme is used, where for each question, the label is chosen as the answer provided by at least two out of the three annotators. These three cases are denoted as **unfiltered data, unanimous, and majority votes** respectively. These experiments were performed on both the baseline model as well as the attention based model.

Further, both the binary classifier and the multi-class classifier baseline models were also analyzed by training with only one of the multimodal inputs, i.e, a model with only the textual question as input with no auxiliary audio input and a model with only the audio signal as input. These are called **question only baseline model** and **audio only baseline model** respectively. This is useful to analyze how well the model captures the information from both the input modalities in predicting the answer word. All the models were trained for 100 epochs with cross-entropy loss and the model with the best validation score is used for evaluation on the test set.

5.4 Results

The results of all the experiments on the Clotho-AQA dataset for binary classification of ‘yes’ or ‘no’ answers are presented in Table 5.1. It can be clearly seen that the model performs better when the answers are unanimous which may indicate the intelligible presence of the answers in the audio compared to the case where different annotators provided different answers. For example, the multimodal baseline model achieves an accuracy of 73.1% when the answers are unanimous compared to 62.7% and 63.2% when the answers are unfiltered and majority votes respectively. As explained in section 4.5, an oracle model can achieve an accuracy of 86.2% on the unfiltered data due to contradicting answers given by the annotators. In the case of unanimous answers and majority voting, an oracle model can achieve 100% accuracy.

The results of single-word answer multiclass classifier experiments on Clotho-AQA_v1 and Clotho-AQA_v2 are summarized in Table 5.2 and Table 5.3 respectively. Since the

Model type	Unfiltered	Unanimous	Majority votes
Audio only baseline	57.5	62.1	58.2
Question only baseline	63.5	71.8	64.4
Multimodal baseline	62.7	73.1	63.2
Multimodal cross attention	66.2	75.4	66.3
Oracle model	86.2	100.0	100.0

Table 5.1. *Accuracies (%) of binary 'yes' or 'no' classifier on Clotho-AQA*

number of unique answer classes is large (828 in Clotho-AQA_v1 and 650 in Clotho-AQA_v2), top-5 and top-10 accuracy scores are also reported.

The multimodal cross-attention model reaches a top-1 accuracy of 57.5% and 61.3% in the case of Clotho-AQA_v1 and Clotho-AQA_v2 respectively. As discussed in section 4.5, an oracle model can achieve a maximum top-1 accuracy of 61% and 65% on Clotho-AQA_v1 and Clotho-AQA_v2 datasets respectively. Since answers were collected from only 3 different annotators, the top-5 and top-10 accuracy of an oracle model can be 100%. It is also clear that after phase II of data cleaning explained in section 4.3, the models perform better on the Clotho-AQA_v2 dataset compared to the Clotho-AQA_v1 dataset.

Model type	Top-1	Top-5	Top-10
Audio only baseline	3.2	13.4	21.1
Question only baseline	55.7	96.8	99.4
Multimodal baseline	54.2	93.7	98.0
Multimodal cross attention	57.5	99.8	99.9
Oracle model	61.0	100.0	100.0

Table 5.2. *Accuracies (%) of single-word answers classifier on Clotho-AQA_v1 dataset.*

Model type	Top-1	Top-5	Top-10
Audio only baseline	4.1	16.8	26.1
Question only baseline	59.1	96.5	99.3
Multimodal baseline	59.8	96.6	99.3
Multimodal cross attention	61.3	99.6	99.9
Oracle model	65.0	100.0	100.0

Table 5.3. *Accuracies (%) of single-word answers classifier on Clotho-AQA_v2 dataset.*

It is evident from the results that the cross-attention mechanism significantly improves the evaluation metrics in both the binary classifier and the multi-class classifier compared to simple feature concatenation used in the baseline model. This supports our initial hypothesis that the attention layer helps the model identify useful features in the audio representation that are closely related to the question posed to give the correct answers.

It is also interesting to note that, in both binary and multi-class classifiers, the uni-modal question only model performs nearly as well compared to the multimodal model which takes in both the audio and question as inputs. This behavior is very common in many VQA tasks such as [36]–[38]. This is due to strong priors that already exists in language models and imbalanced dataset. For example, for questions which has the word 'animal' in them, 'dog' is the most common answer in the dataset. Similarly, for questions with the word 'chirp', 'bird' is the most common answer. The models learn these strong biases from these imbalanced data and hence ignore the audio inputs while predicting some of the answers.

6. CONCLUSION AND FUTURE WORK

Audio question answering is a multimodal translation task in which a system analyses an audio signal and a natural language question related to it as inputs and provides a natural language answer as its output. A new dataset called Clotho-AQA with audio signals, questions, and corresponding answers was created and neural network models were designed to accomplish the audio question answering task on this dataset.

Clotho-AQA is an audio question answering dataset consisting of 1991 audio files selected from the Clotho dataset. For each audio file, six questions were collected from annotators living in native English-speaking countries by crowdsourcing using amazon mechanical turk. The questions were collected such that it is possible to answer each question with a single word or ‘yes’ or ‘no’. The answers to these questions were also collected by another round of crowdsourcing using AMT. For each question, the answers were collected from three different annotators independently. We then post-processed the collected answers to remove unique words and replace them with commonly occurring suitable answers.

As a baseline model, we trained a binary classifier for ‘yes’ and ‘no’ answers and a multi-class classifier for single-word answers. The baseline models use pre-trained feature extractors for both audio and question inputs. The extracted audio and textual features were concatenated and passed to a classifier to produce the model predictions. The baseline binary classifier achieved an accuracy of 62.7% on unfiltered data and the baseline multi-class classifier achieved a top-1 accuracy of 54.2%.

We also proposed attention-based architecture for audio question answering task. Here the extracted audio and textual features are passed through cross-attention layers to learn relationships between the two modes. Our results clearly proved that the cross-attention mechanism helps the model to learn better relationships between the input question and the audio compared to the baseline. The attention-based model produced an accuracy of 66.2% for the binary classification task. Similarly, the top-1 accuracy for multi-class classification increased to 57.5% for the attention model.

Further, we discussed in detail some of the issues present in the Clotho-AQA dataset called Clotho-AQA_v1 in this work. These include different specificity of answers, singular and plural forms of the same answer words, and different tenses of the same answer words. We fixed these issues and provided a polished version of this dataset denoted as Clotho-AQA_v2. Our proposed baseline and attention-based models performed comparatively better on the new dataset. The baseline multi-class classifier model achieved a top-1 accuracy of 59.8% and the attention-based model achieved a top-1 accuracy of 61.3% on the Clotho-AQA_v2 dataset.

In the future, we plan to extend the dataset to include variable-length answers. We also intend to reduce data imbalance and minimize language model biases in the new dataset. We also plan to build sophisticated model architectures that can take full advantage of the multimodal inputs for audio question answering tasks.

REFERENCES

[1] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “Squad: 100,000+ questions for machine comprehension of text,” in *Empirical Methods in Natural Language Processing*, 2016.

[2] A. Trischler, T. Wang, X. Yuan, *et al.*, “Newsqa: A machine comprehension dataset,” in *Proceedings of the 2nd Workshop on Representation Learning for NLP*, 2017.

[3] M. Malinowski and M. Fritz, “A multi-world approach to question answering about real-world scenes based on uncertain input,” in *Neural Information Processing Systems*, 2014.

[4] Y. Zhu, O. Groth, M. S. Bernstein, and L. Fei-Fei, “Visual7w: Grounded question answering in images,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] A. Agrawal, J. Lu, S. Antol, *et al.*, “VQA: Visual question answering,” *International Journal of Computer Vision*, vol. 123, pp. 4–31, 2015.

[6] K. Kafle and C. Kanan, “Visual question answering: Datasets, algorithms, and future challenges,” *Computer Vision and Image Understanding*, vol. 163, pp. 3–20, 2017.

[7] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu, “Are you talking to a machine? dataset and methods for multilingual image question,” in *Neural Information Processing Systems*, 2015.

[8] J. Lei, L. Yu, M. Bansal, and T. L. Berg, “TVQA: Localized, compositional video question answering,” in *Empirical Methods in Natural Language Processing*, 2018.

[9] D. Patel, R. Parikh, and Y. Shastri, “Recent advances in video question answering: A review of datasets and methods,” in *International Conference on Pattern Recognition*, 2020.

[10] K.-m. Kim, M.-O. Heo, S. Choi, and B.-T. Zhang, “Deepstory: Video story QA by deep embedded memory networks,” *International Joint Conference on Artificial Intelligence*, 2017.

[11] Z. Yu, D. Xu, J. Yu, *et al.*, “Activitynet-qa: A dataset for understanding complex web videos via question answering,” *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[12] J. Abdelnour, G. Salvi, and J. Rouat, *Clear: A dataset for compositional language and elementary acoustic reasoning*, Neural Information Processing Systems, 2018.

[13] H. M. Fayek and J. Johnson, “Temporal reasoning via audio question answering,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020.

[14] K. Drossos, S. Lipping, and T. Virtanen, “Clotho: An audio captioning dataset,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[15] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, 2017.

[17] S. Lipping, P. Sudarsanam, K. Drossos, and T. Virtanen, “Clotho-aqa: A crowdsourced dataset for audio question answering,” in *30th European Signal Processing Conference (EUSIPCO)*, 2022.

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space", *International Conference on Learning Representations*, 2013.

[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in Neural Information Processing Systems*, 2013.

[20] D. E. Rumelhart and J. L. McClelland, “Learning internal representations by error propagation,” in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*. 1987, pp. 318–362.

[21] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches", *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, 2014.

[22] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate", *International Conference on Learning Representations*, 2015.

[23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *International Conference on Learning Representations*, 2021.

[24] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, “Convolution-augmented transformer for semisupervised sound event detection,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, 2020.

[25] P. Sudarsanam, A. Politis, and K. Drossos, "Assessment of Self-Attention on Learned Features For Sound Event Localization and Detection", *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2021.

[26] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, “Clevr: A diagnostic dataset for compositional language and elementary visual reasoning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[27] F. Font, G. Roma, and X. Serra, “Freesound technical demo,” in *ACM International Conference on Multimedia*, ACM, Barcelona, Spain: ACM, Oct. 2013.

[28] M. Ren, R. Kiros, and R. Zemel, “Exploring models and data for image question answering,” in *Advances in Neural Information Processing Systems*, 2015.

[29] R. Krishna, Y. Zhu, O. Groth, *et al.*, “Visual genome: Connecting language and vision using crowdsourced dense image annotations,” *arXiv preprint arXiv:1602.07332*, 2016.

[30] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, “Microsoft coco: Common objects in context,” in *European Conference on Computer Vision (ECCV)*, 2014.

[31] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *International Conference on Learning Representations, (ICLR)*, 2015.

[32] J. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, “Look, listen, and learn more: Design choices for deep audio embeddings,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[33] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, “Advances in pre-training distributed word representations,” in *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[34] R. Arandjelovic and A. Zisserman, “Look, listen and learn,” *IEEE International Conference on Computer Vision (ICCV), Venice, Italy*, Oct. 2017.

[35] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, *et al.*, “Audio set: An ontology and human-labeled dataset for audio events,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing 2017*, 2017.

[36] J. Devlin, S. Gupta, R. Girshick, M. Mitchell, and C. L. Zitnick, “Exploring nearest neighbor approaches for image captioning,” *arXiv preprint arXiv:1505.04467*, 2015.

[37] P. Zhang, Y. Goyal, D. Summers-Stay, D. Batra, and D. Parikh, “Yin and yang: Balancing and answering binary visual questions,” *in Computer Vision and Pattern Recognition (CVPR)*, 2016.

[38] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus, “Simple baseline for visual question answering,” *arXiv preprint arXiv:1512.02167*, 2015.