Temporally Aligning Long Audio Interviews with Questions: A Case Study in Multimodal Data Integration

Piyush Singh Pasi[1] , **Karthikeya Battepati**[1] , **Preethi Jyothi**[1] , **Ganesh Ramakrishnan**[1] , **Tanmay Mahapatra**[2] , **Manoj Singh**[2]
[1]Indian Institute of Technology, Bombay
[2]CARE India Solutions for Sustainable Development
{piyushsinghpasi, karthikeyabatte, pjyothi, ganesh}@cse.iitb.ac.in, {tanmay, manojks}@careindia.org

Abstract

The problem of audio-to-text alignment has seen significant amount of research using complete supervision during training. However, this is typically not in the context of long audio recordings wherein the text being queried does not appear verbatim within the audio file. This work is a collaboration with a non-governmental organization called CARE India that collects long audio health surveys from young mothers residing in rural parts of Bihar, India. Given a question drawn from a questionnaire that is used to guide these surveys, we aim to locate where the question is asked within a long audio recording. This is of great value to African and Asian organizations that would otherwise have to painstakingly go through long and noisy audio recordings to locate questions (and answers) of interest. Our proposed framework, INDENT, uses a cross-attention-based model and prior information on the temporal ordering of sentences to learn speech embeddings that capture the semantics of the underlying spoken text. These learnt embeddings are used to retrieve the corresponding audio segment based on text queries at inference time. We empirically demonstrate the significant effectiveness (improvement in R-avg of about 3%) of our model over those obtained using text-based heuristics. We also show how noisy ASR, generated using state-of-the-art ASR models for Indian languages, yields better results when used in place of speech. INDENT, trained only on Hindi data is able to cater to all languages supported by the (semantically) shared text space. We illustrate this empirically on 11 Indic languages.

1 Introduction

Audio surveys and oral interviews are routinely used as a means for data collection in many parts of the world [Jones, 2003], [Reichmann *et al.*, 2010]. Apart from ease of use, audio surveys are also very inclusive since they naturally allow for data to be collected from illiterate or physically challenged individuals [Heinritz *et al.*, 2022]. Several audio surveys by governmental and non-governmental organizations are conducted with the aim of improving social outcomes such as health and education. These surveys are typically accompanied by predefined questionnaires that guide the interviewer. Temporally aligning questions to where each question was asked in an interview becomes very important for organizations collecting the data, so as to quickly retrieve answers to relevant questions from long audio files.

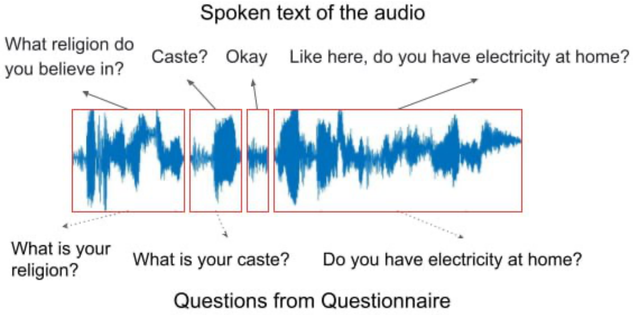
Spoken text of the audio
What religion do you believe in?
Caste?
Okay
Like here, do you have electricity at home?
What is your religion?
What is your caste?
Do you have electricity at home?
Questions from Questionnaire

Figure 1: Illustration of speech in audio and relevant questions

In this work, we use long audio health surveys collected from mothers with young children residing in rural areas of Bihar, India. This data was collected by a non-governmental organization CARE India [CARE, 2023] as part of their larger effort towards improving maternal and neonatal health. The interviews are typically long, spanning more than 40 minutes on average, and are based on a fixed questionnaire designed by CARE. Given a written question drawn from the questionnaire, our main goal is to extract a short audio snippet from the long recording that matches the given question. This task is of utmost importance to organizations such as CARE who would otherwise have to resort to tedious manual searches through these long audio recordings to retrieve answers to various questions.

Most prior work on cross-modal segment retrieval relies on strong supervision in the form of temporally-aligned text annotations [Anne Hendricks *et al.*, 2017], [Gao *et al.*, 2017], [Liu *et al.*, 2018], [Xu *et al.*, 2018], [Gupta *et al.*, 2021], [Javed *et al.*, 2022]. For long audio files, creating such time-aligned annotations using human annotators would require significant amount of time and money. Since we are working with audio surveys derived from a fixed sequence of questions in a questionnaire, we use a weaker form of supervision involving larger audio segments paired with sets of temporally ordered questions. We present a new architecture INDENT[1], that is trained using such weak supervision of larger audio segments and sets of questions. INDENT uses an attention-based mechanism and is trained with contrastive losses across audio-text modalities to ground each question in the larger segment with its corresponding audio.

Our task of audio segment retrieval using textual queries is particularly challenging for the following reasons: 1. The audio surveys with the young mothers are conducted in noisy home environments, thus resulting in very noisy audio recordings. 2. Questions from the questionnaire are not asked verbatim and are typically paraphrased by the interviewer when posed to the mothers. Figure 1 illustrates this phenomenon. 3. While the questions appear in formal text in the official language of Bihar, *i.e.*, Hindi, the audio recordings could contain variations that are typical of the local dialects spoken by the interviewees.

To summarize, our main contributions are: (i) We tackle a high-utility problem of locating short audio segments from within long audio surveys based on textual queries. These audio surveys are collected by CARE India in real environments and present many interesting challenges that we outline above. The dataset is enriched with fine-grained annotations on the evaluation splits and coarse-grained annotations on the training data that is sufficient to enable weak supervision. (ii) We design INDENT, a new model for text-to-audio retrieval from within long audio surveys given a question in text. INDENT is trained using weak supervision consisting of longer audio segments and sets of temporally ordered questions.

The problem of question-to-audio retrieval can also serve as an intermediate step for various downstream tasks such as audio-driven form filling and audio-driven question answering, by extracting answers from the survey recordings once the relevant question is isolated. We leave such extensions to future work and focus on the challenging problem of retrieving audio segments from long audios based on questions in text.

2 Related Work

Most relevant to our work are (i) approaches that focus on temporal sentence grounding in videos, where the task is to retrieve a video that semantically corresponds to a query in natural language, and (ii) audio text alignment for Automatic Speech Recognition (ASR).

2.1 Video Moment Retrieval Using Text Queries

Prior work has explored various approaches for the task of temporally grounding captions or descriptions in videos in both fully and weakly supervised settings [Zhang *et al.*, 2022]. Some of the prominent approaches are discussed below. Moment Context Networks (MCN) [Anne Hendricks *et al.*, 2017] learn a shared embedding for video temporal context features (*i.e.*, video features integrated with temporal endpoint features that indicate when a moment occurs in a video) and LSTM-based language features. The CTRL architecture [Gao *et al.*, 2017] constitutes a temporal regression network to produce alignment scores and location offsets using combined representations for visual and text domains. A language-temporal attention network has also been proposed [Liu *et al.*, 2018] to learn word attention based on the temporal context information in the video. Temporal boundary annotations for sentences have been used to train a segment proposal network using attention [Xu *et al.*, 2018]. All these approaches assume the fully supervised setting. A Text-Guided Attention (TGA) model was proposed for video moment retrieval using text queries in a weakly supervised setting [Mithun *et al.*, 2019]. However, TGA does not utilize the temporal order of sentences unlike our approach, INDENT that employs a Gaussian-weighted attention to leverage this temporal information.

2.2 Audio-Text Alignment For Speech Recognition

Most of the widely used ASR models that generate the best transcripts for various languages require large amounts of labeled transcripts for training [Chadha *et al.*, 2022; Javed *et al.*, 2022]. Methods proposed for low resource settings [Anguera *et al.*, 2014] also expect access to some annotated transcripts during training, which are not available in our setting. Models that use CTC or sequence-to-sequence loss functions [Synnaeve *et al.*, 2019; López and Luque, 2022] require audio files with their corresponding transcripts to be used during training. Such models cannot be employed with our CARE India dataset, since we do not have any transcribed speech and the questions asked in the audio recordings are only semantically similar to (and often, paraphrases of) the questions listed in the questionnaire.

Recently, a few methods have been proposed for speech recognition using unaligned speech and text [Ao *et al.*, 2021; Chen *et al.*, 2022] and generating speech embeddings that capture the semantics of the underlying spoken text by aligning both text and speech embedding spaces using unsupervised techniques [Chung *et al.*, 2018; Chen *et al.*, 2018]. These models require word-level audio segments for semantically rich speech embeddings, which are generally not easily available or extractable. Other prior work require very large amounts of unpaired speech and text and do not currently serve many Indian languages [Ao *et al.*, 2021; Chen *et al.*, 2022].

3 CARE India Dataset

We create a new "CARE India Dataset" that contains speech interviews conducted by the CARE India Organization[2]. These interviews focus on analyzing maternal health and empowering women in rural areas in the state of Bihar, India. Each interview is approximately 40 minutes long and is recorded in the household premises of the interviewee, thus yielding very noisy recordings. The interviewer refers to a questionnaire as a guide for the duration of the interview. Searching for a specific question from the questionnaire within the noisy long audio recording is not the only challenge. Questions are not asked verbatim during the course

[1] INDENT can be expanded as "al**I**gning lo**N**g au**D**io int**E**rviews a**N**d ques**T**ions".

[2] https://bihar.care.org/

of the conversation and could be paraphrased. The questions are also drawn from a diverse pool including multiple-choice type, numerical and subjective questions. All these factors add to the overall difficulty of isolating the correct audio segment that maps to a given question. CARE India (and other similar organizations) could greatly benefit from an automated solution that given a question in text can identify roughly where it appeared in a long audio recording. Without such a tool, volunteers would have to tediously go through long and noisy audio recordings to search for where a question appeared in it. This work is our first step towards building such a tool. We present a new model INDENT that can be trained on longer audio segments spanning multiple questions and identify potential audio segments within a test audio recording that could be matched with a test query.

Each interview in the train set is annotated at the segment-level. A segment contains m consecutive questions and is of variable duration. Segments do not overlap or share questions. If a question begins within a segment, it would also end in that segment. We choose to set $m = 5$, since this granularity gives us segments that are not too long (which would make learning difficult) and not too short (which would otherwise overwhelm the human annotators). For each audio segment containing five questions, we annotate the start of the first question and the end of the fifth question. Our development and test sets have fine-grained annotations with start and end timestamps for each question. However, such a per-question annotation takes much longer. We find that annotating train files takes around 4 hours per audio recording, while it takes more than 6 hours to annotate test files from a similarly-sized audio recording. Detailed dataset statistics for our train, development and test sets are presented in Table 1.

TRAIN SET	
Total no. of interviews	34
Total no. of segments	1223
Avg. no. of segments in an interview	35.97
Avg. segment duration (sec)	54.64
Avg. no. of chunks per segment	14.31
Avg. chunk duration (sec)	1.7
Avg. interview duration† (min)	42.1
Total Train set duration† (hour)	23.85
Total no. of questions asked	6115
Avg. no. of questions asked in an interview	179.85
DEV SET	
Total no. of interviews	3
No. of chunks in an interview	634.33
Avg. interview duration† (min)	34.2
Total Dev set duration† (hour)	1.71
Avg. chunk duration (sec)	2.04
Total no. of questions asked	538
Avg. no. of questions asked in an interview	179.33
Avg. question duration (sec)	2.87
TEST SET	
Total no. of interviews	5
No. of chunks in an interview	603.4
Avg. interview duration† (min)	36.28
Total Test set duration† (hour)	3.02
Avg. chunk duration (sec)	1.96
Total no. of questions asked	629
Avg. no. of questions in an interview	125.8
Avg. question duration (sec)	1.78
No. of question in fixed questionnaire Q	1555

Table 1: CARE India Dataset Statistics, † denotes duration without any preprocessing

4 INDENT: Architecture and Methodology

We formally define our problem as follows. Given a question q drawn from a fixed questionnaire Q and an audio interview f within which q is asked, we want to retrieve the audio segment $s \in f$ that exactly matches q. On an average, f is more than 40 minutes long. In Table 1, we provide more detailed statistics on the data distribution. As described in Section 3, during training, we assume access to longer audio segments containing $m = 5$ questions. These segmentations are manually annotated. At test time, we extract non-overlapping audio segments of fixed duration from the test audio interview f and identify the segment that is the best match for a given question using our trained INDENT.

Overview of Workflow. INDENT consists of three main components: 1. Speech Encoder 2. Question Encoder 3. Gaussian-weighted Cross-attention Network. The speech encoder consists of audio preprocessing and feature extractor modules. We use an audio preprocessing module to remove noise from the speech interviews and break segments into smaller chunks of roughly 2 seconds duration each. We extract speech features at the chunk-level. The question encoder extracts question features at the sentence-level. Hence, the time resolution of the audio chunk features and the question features are very different (*c.f.* Figure 1). Using both audio chunk and question feature sequences as inputs, we try to align chunks with questions and simultaneously bridge the modality gap between speech and text in a shared space using a Gaussian Weighted Cross Attention module and contrastive learning. At inference, for a speech interview, we rank segments using the dot-product scores of the chunks and the given question. In Figure 2, we present a schematic diagram of the overall architecture of INDENT. Next, we elaborate on each of the model components.

4.1 Speech Encoder

The audio recordings contain speech both from the interviewer and the interviewee; the latter is typically very faint and more difficult to isolate. As mentioned in Section 3, the audio recordings are annotated with start and end times of large segments containing $m = 5$ questions each. Within each of these segments, we are only interested in parts corresponding to the interviewer. To isolate these parts, we attempted several techniques including voice activity detection (VAD) [Wiseman, 2017], speaker diarization [Bredin *et al.*, 2020] and source separation techniques [Ravanelli *et al.*, 2021]. After qualitatively analyzing the respective outputs, we found VAD to yield the best quality outputs. Using VAD, we split each annotated segment s into n_s chunks $\{c_i^s\}_{i=1}^{n_s}$. This results in a varying number of chunks across segments. We note here that the number of chunks n_s can be higher than the number of questions m in a segment s. The microphone often fails to adequately capture the interviewee's voice, and we therefore assume (after qualitative validation) that the chunks after VAD would primarily contain the interviewer's speech.

Once we have the chunk sequence $\{c_i^s\}_{i=1}^{n_s}$, we use a frozen pretrained speech encoder (wav2vec2.0 [Baevski *et al.*, 2020]) to extract features for chunks $\{\boldsymbol{c}_i^s\}_{i=1}^{n_s}$ where $\boldsymbol{c}_i^s \in R^{T_i \times d'}$, T_i is the number of time frames for chunk c_i^s. The speech encoder produces a d'-dimensional feature vector for each speech frame in c_i^s. To reduce $\boldsymbol{c}_i^s$ to one aggregate feature vector per chunk, we apply a 1D convolution and take the mean across all T_i features. Next, we project these chunk features to a d-dimensional shared space using a linear layer. We then apply self-attention over the segment so that each chunk can gather context from other chunks. We also add skip connections wherever appropriate. The left block in Figure 2 represents the speech encoder with layers and skip connections. More formally:

$$\begin{aligned}\{\hat{\boldsymbol{c}}_i^s\}_{i=1}^{n_s} &= \text{MeanPool}(\text{Conv1D}\,(\{\boldsymbol{c}_i^s\}_{i=1}^{n_s}))\\ C^s &= \{C_i^s\}_{i=1}^{n_s} = \text{Linear}(\{\hat{\boldsymbol{c}}_i^s\}_{i=1}^{n_s})\\ \boldsymbol{C}^s &= \{\boldsymbol{C}_i^s\}_{i=1}^{n_s} = \text{SelfAttn}(C^s, C^s, C^s)\end{aligned}$$

where $C_i^s \in \mathbb{R}^d$ is the chunk feature after a Linear layer, $\text{SelfAttn}(k, q, v) = \text{softmax}(\frac{k.q^\intercal}{\sqrt{d}})v$, $q, k, v \in \mathbb{R}^{n_s \times d}$ and $\boldsymbol{C}_i^s$ is the chunk feature after self-attention.

4.2 Question Encoder

We use a frozen pretrained sentence transformer network to extract a d-dimensional feature vector. During training, for each segment s we have m questions, and hence we compute $\boldsymbol{q}^s = [\boldsymbol{q}_1^s, \cdots, \boldsymbol{q}_m^s]$ where $\boldsymbol{q}_j^s \in \mathbb{R}^d$. We opt for sentence-level features for textual questions since the spoken and textual questions are semantically similar at the sentence-level but need not have any correspondence at the word-level (*e.g.*, when the spoken question is a paraphrase of the written question). Thus, extracting word-level features from the textual questions can lead to sub-optimal alignments. We also note here that we are doing weakly supervised training with no access to temporal boundaries but only the order of the occurrence of questions within a segment. This motivates our choice of using sentence-level features. The right block in Figure 2 shows the question encoder. It is important to keep the question encoder frozen since the pretrained sentence transformer is trained with massive amounts of text data in comparison to our training data. We only want to learn effective semantic alignments from the spoken utterances to the already pretrained sentence embeddings.

For each segment s, we now have speech chunk features $\boldsymbol{C}^s$ and textual features from the questions $\boldsymbol{q}^s$ and we need to learn a cross-modal alignment between these two feature sequences. Towards this, we propose a Gaussian-weighted Cross Attention scheme.

4.3 Gaussian-weighted Cross Attention

Given $\boldsymbol{C}^s$ and $\boldsymbol{q}^s$, we aim to learn an alignment between $\boldsymbol{q}^s$, representing m sentence embeddings and $\boldsymbol{C}^s$, representing n_s chunk embeddings. We propose a Gaussian-weighted cross attention module with a contrastive learning objective [Chen *et al.*, 2020] in order to learn this alignment. Typically, $n_s > m$ implying that a question spans more than one speech chunk. To align a chunk $\boldsymbol{C}_i^s$ with a question $\boldsymbol{q}_i^s$, we consider an "anchor chunk" feature $\hat{\boldsymbol{C}}_i^s$ using $\boldsymbol{q}^s$. While chunk is anchored to one question, the neighboring questions can also provide context. We assume some consecutive chunks $\boldsymbol{C}_{a:b}^s$ combine to form the question $\boldsymbol{q}_i^s$, but we do not know the temporal boundaries of $\boldsymbol{q}_i^s$ (given our weakly-supervised setting). Further, speech features $\boldsymbol{C}_i^s$ are not semantically rich to guide us with any weak boundaries. However, we know that each chunk unambiguously belongs to a single underlying question. Hence, we represent each chunk $\boldsymbol{C}_i^s$ as a linear combination of $\boldsymbol{q}^s$ but anchor the chunk $\boldsymbol{C}_i^s$ to a single $\boldsymbol{q}_i^s$ by making one of the weights high and the rest much smaller to accumulate some neighboring context. A Gaussian distribution with a moving mean and standard deviation serves our purpose.

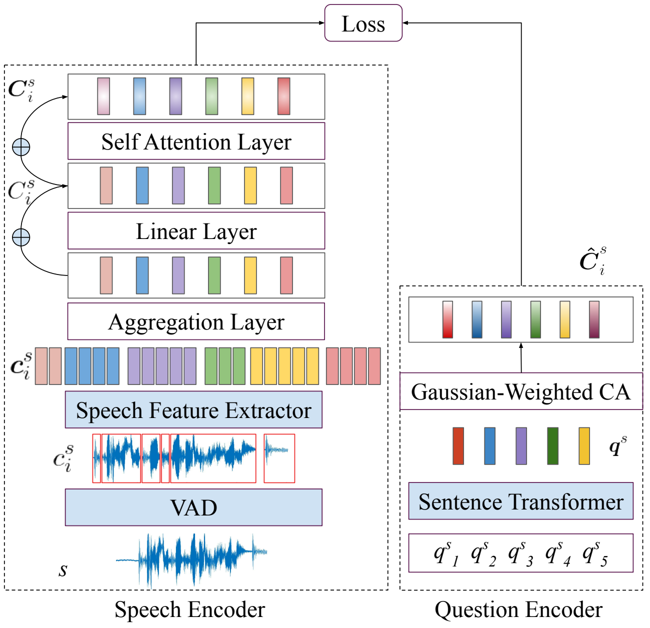

Figure 2: End-to-End INDENT Architecture. Modules filled with blue color are frozen.

Start-of-segment chunks and end-of-segment chunks map to questions $\boldsymbol{q}_1^s$ and $\boldsymbol{q}_m^s$, respectively, with high probability. Hence, pivoting on n_s, we move the Gaussian mean and vary the standard deviation as a function of the position of the chunk within the segment:

$$\mu_i = \frac{(i-1)(m-1)}{n_s - 1} \tag{1}$$

$$\sigma_i = \alpha \cdot \min(i-1, n_s - i) \tag{2}$$

$$\mathcal{G}_i = \text{Gaussian}(\mu_i, \sigma_i) \tag{3}$$

where μ_i and σ_i are the mean and standard deviation used to generate the Gaussian distribution $\mathcal{G}_i$ for chunk c_i^s and α is a scaling factor. From Eqn 3, we see that for each position i, the mean shifts by $\frac{m-1}{n_s-1}$ from the previously calculated mean.

We sample values $\{a_j^i\}_{j=1}^m$ from $\mathcal{G}_i$ as shown in Fig. 3, apply min-max normalization and generate an anchor chunk $\hat{\boldsymbol{C}}_i^s$ using $\boldsymbol{q}^s$ for each chunk $\boldsymbol{C}_i^s$. Hence, $\{a_j^i\}_{j=1}^m$ has a peak value a_j^i and also has some weight allocated to the other questions. Thus, we generate a representation corresponding to each audio chunk $\boldsymbol{C}_i^s$ in the shared semantic space as $\hat{\boldsymbol{C}}_i^s = \sum_{j=1}^m a_j^i q_j{}^s$. Since $\hat{\boldsymbol{C}}_i^s$ and $\boldsymbol{C}_i^s$ are both chunk-level representations, we apply a contrastive objective between them to bridge the modality gap. Since the question encoder is frozen, contrastive learning facilitates the speech encoder to better align with the text-based sentence embeddings.

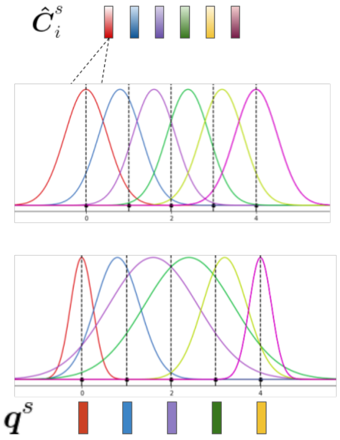

Figure 3: Gaussian-weighted Cross Attention (CA). Gaussian distribution of weights (a_j^i) for fixed and varying σ cases. We sample at weights at each black-dot $(0, \cdots, m-1)$ and the weight value is intersection of the Gaussian distribution and the dotted line which results in very peaky distribution for peripheral chunks & vice versa.

Dynamic In-Audio Negative Sampling. For contrastive learning, we need negative chunks. Specifically, negative chunks from the same interview are desirable to prevent the model from learning easy solutions by merely exploiting speaker characteristics. During training, we create groups by randomly shuffling all segments in the speech interview and making groups of n segments. If the number of segments in the audio interview is not divisible by n, we also create a group with the last n segments in order to ensure that all the segments are covered during training. For each chunk in segment s_x, the corresponding negative chunks will be from segment s_y $(x \neq y)$ in the same group g. Chunks from all such s_y are potential negatives for each chunk of s_x. We denote the number of all potential negatives for s_x by k_x. To keep the number of negatives for each chunk constant inside a batch, we first compute the smallest k_x across s_x's in a batch and randomly sample those many chunks as negative examples for each chunk in the batch. We also explore data augmentation by performing reshuffling of all segments to create more new groups. The number of times we perform data augmentation is denoted by the hyperparameter D. Reshuffling is also performed across epochs.

Training Objective. We define the following contrastive loss [Chen *et al.*, 2020] objective:

$$L_{\text{st}} = \frac{-1}{|C|} \sum_{C} \log \left(\frac{\exp(\boldsymbol{C}_i^{s\top} \hat{\boldsymbol{C}}_i{}^{s})}{\sum_{j \in n_g} \exp(\boldsymbol{C}_j^{s\top} \hat{\boldsymbol{C}}_i{}^{s'})} \right)$$

$$L_{\text{ss}} = \frac{-1}{|C|} \sum_{C} \log \left(\frac{\exp(\boldsymbol{C}_i^{s\top} \boldsymbol{C}_i{}^{s})}{\sum_{j \in n_g} \exp(\boldsymbol{C}_j^{s\top} \boldsymbol{C}_i{}^{s'})} \right)$$

where n_g is a union of i and dynamically sampled negative chunk indices (as described earlier), $s, s' \in g, s = s'$ if $j = i$, C denotes all chunks in a batch; L_{st} aligns the speech and text spaces while L_{ss} regularizes the network by explicitly encouraging negative speech chunks $\boldsymbol{C}_j^s$ $(j \in n_g, j \neq i)$ to be far apart from $\boldsymbol{C}_i^s$ in the shared space. INDENT is optimized over the loss $L = L_{\text{st}} + L_{\text{ss}}$. We apply the label smoothing technique [Reed *et al.*, 2014] that forces INDENT to lower the confidence of the positive chunk by redistributing a small fraction of its probability mass uniformly to all the negative chunks. (In the experiments, we set positive probability to be 0.95 and redistribute 0.05 probability mass evenly among all the negatives.)

5 Experiments and Results

5.1 Experimental Setup

Question Encoder. We use pretrained Language-agnostic BERT Sentence Embedding (LaBSE) [Feng *et al.*, 2022] as sentence embeddings. For each question q, LaBSE produces a 768-dimensional embedding. Since we keep the question encoder frozen, a helpful consequence of using LaBSE is that we can use questions in any language that are well-aligned in the LaBSE embedding space. In Table 2, we present experiments on questions in multiple Indian languages.

Speech Encoder. We use a pretrained voice activity detector (VAD) [Wiseman, 2017] to split segment s into n_s chunks, $\{c_1^s, \cdots, c_{n_s}^s\}$. To extract the chunkwise speech features $\boldsymbol{c_i}^s$, we use a state-of-the-art multilingual model IndicWav2vec-Hindi [Javed *et al.*, 2022]. Each chunkwise speech vector is a 1024-dimensional feature vector. To aggregate, we apply a convolution layer with 1 filter per channel (*i.e.*, a total of 1024 filters) of receptive-field 20 and stride 2, followed by GELU activation [Hendrycks and Gimpel, 2016] and dropout and a final mean pooling on $\boldsymbol{c}_i^s$ to derive $\hat{\boldsymbol{c}}_i^s$. We then linearly project $\hat{\boldsymbol{c}}_i^s$ to a 768-dimensional vector to generate $C_i^s \in C^s$. We subsequently apply self attention on C^s to generate $\boldsymbol{C}^s$. We present experiments with different speech feature extractors in Section 5.2.

Training. We train INDENT end-to-end keeping VAD [Wiseman, 2017], Question Encoder LaBSE [Feng *et al.*, 2022] and speech feature extractor IndicWav2Vec-Hindi [Javed *et al.*, 2022] modules frozen using the loss defined in Section 4.3. We use the Adam optimizer with a learning rate 3e-4, a step scheduler with step-size 10 and decay factor gamma 0.1. After hyperparameter tuning (described in Section 5.2), we set batch size $|B| = 4, n = 4, D = 2, \sigma = 0.5$ and train for 40 epochs.

Model Variants. We refer to the complete model with the speech encoder and question encoder, trained using contrastive losses, as INDENT. To compare how INDENT performs with text instead of speech, we train INDENT-T by replacing the speech feature extractor in the speech encoder with ASR predictions for which we extracted LaBSE embeddings. NO-TRAIN is a learning-free technique that computes a simple matching based on dot-products between LaBSE embeddings of questions with LaBSE embeddings of ASR predictions from the speech.

Evaluation & Metrics. During inference, we create fixed-chunk-size segments since we do not have segment boundaries during inference. We need segment information for local context; removing self-attention within a segment deteriorates the performance significantly. We set segment-size to 14 (average number of chunks in the train set). We compute dot-product score $q^\intercal \boldsymbol{C}_i^s$ for the given question q across all chunks $\boldsymbol{C}_i^s$. We rank each segment s using $\max_{i=1\cdots n_s}(q^\intercal \boldsymbol{C}_i^s)$ (higher score is better). We evaluate using the recall@k (R@k) metric. If the ground truth segment s of the question q appears in the top k highest scoring segments, the algorithm is considered to have retrieved the correct segment for R@k. Our reported R@k numbers are a mean across interviews of the ratio of the number of questions correctly retrieved to the total number of questions asked $\left(\frac{\text{no. of correctly retrieved questions}}{\text{total no. of questions}}\right)$. We report R@1, R@5, R@10 and average of the three is R-avg.

5.2 Results & Analysis

In Table 3, we present our main results. INDENT outperforms NO-TRAIN by an absolute 6.4% indicating that our training method has successfully aligned speech features to the semantic space of LaBSE embeddings [Feng *et al.*, 2022]. INDENT is able to exploit very small amounts of weakly-annotated data to successfully bridge the modality gap and generate semantically-rich speech features. We also train INDENT-T using ASR transcripts instead of speech using our architecture and we see large (absolute) 26.4% and 19.98% gains in R-avg performance compared to NO-TRAIN and INDENT, respectively. This suggests: 1. The quality of ASR transcripts is reasonably good with WER 17.8 for the base model without any LM (as reported by [Javed *et al.*, 2022]). 2. Aligning across speech and text modalities with small amounts of weakly labeled data is a challenging task. Bootstrapping using jointly trained speech and text models might be worth exploring as future work [Ao *et al.*, 2021; Chen *et al.*, 2022]. Our model INDENT still performs better than NO-TRAIN and can be quite effective when used with low-resource languages that do not have good ASR. We also perform experiments with other speech feature extractors such as Vakyansh [Chadha *et al.*, 2022] and XLSR [Conneau *et al.*, 2020], but both underperform compared to IndicWav2vec-Hindi [Javed *et al.*, 2022].

Feature extractor	Model Variant	R@1	R@5	R@10	R-avg
NO-TRAIN		19.8	38.5	51.6	36.7
XLSR [Conneau *et al.*, 2020]	INDENT	9.3	30.5	43.8	27.9
Vakyansh [Chadha *et al.*, 2022]	INDENT-T	35.6	62.9	74.9	57.8
	INDENT	15.0	34.9	47.9	32.6
IndicASR [Javed *et al.*, 2022]	INDENT-T	40.9	69.0	79.4	63.1
	INDENT	21.1	46.5	61.8	43.1

Table 3: Main test results. We experiment with different feature extractors and model variants.

A useful consequence of training INDENT using a frozen question encoder is that we can query our model with any of the 109 languages supported by LaBSE. We evaluate INDENT on 11 different Indian languages unseen during training. During inference, we translate the questionnaire using IndicTrans's Indic2English (for Hindi to English translation) and Indic2Indic (for Hindi to other Indic language translation) models [Ramesh *et al.*, 2022]. Note that during training we use speech in Hindi matched to questions in Hindi, while during inference we query using text in multiple Indian languages. From the results shown in Table 2, we observe that INDENT outperforms NO-TRAIN for all Indian languages. INDENT-T performs the best, thus pointing to high-quality ASR transcriptions. Interestingly, performance on English is better with NO-TRAIN compared to INDENT. This could be because LaBSE was trained predominantly on English data.

Tables 4, 5, 6, and 7 show results on the dev set using INDENT from various ablation experiments by tuning important

Language	NO-TRAIN				INDENT				INDENT-T			
	R@1	R@5	R@10	R-avg	R@1	R@5	R@10	R-avg	R@1	R@5	R@10	R-avg
Hindi (hi)	19.8	38.5	51.6	36.7	**21.1**	**46.5**	**61.8**	**43.1**	**40.9**	**69.0**	**79.4**	**63.1**
Assamese (as)	17.7	36.0	50.3	34.7	16.4	39.4	54.3	36.7	33.8	62.8	74.6	57.1
Bengali (bn)	18.3	37.9	51.7	36.0	20.3	45.4	57.9	41.2	37.2	67.7	78.6	61.2
English (en)	**21.5**	**43.2**	**55.9**	**40.2**	16.9	40.4	53.7	37.0	37.2	64.5	78.0	59.9
Gujarati (gu)	19.2	38.5	50.9	36.2	20.0	45.1	60.3	41.8	38.7	68.6	78.8	62.0
Kannada (kn)	19.1	37.1	52.8	36.3	18.7	45.0	58.6	40.8	39.2	66.6	78.1	61.3
Malayalam (ml)	18.5	35.1	49.1	34.2	18.3	44.7	57.8	40.3	35.0	65.2	77.4	59.2
Marathi (mr)	19.0	36.2	52.3	35.8	18.7	44.8	59.6	41.0	38.4	65.7	77.0	60.4
Oriya (or)	20.3	39.6	52.4	37.4	18.9	44.4	58.4	40.5	39.0	67.0	78.0	61.4
Punjabi (pa)	19.5	37.9	50.5	36.0	20.3	45.7	58.5	41.5	39.2	67.4	78.3	61.6
Tamil (ta)	18.9	38.5	52.0	36.4	17.7	43.8	59.0	40.2	34.9	66.5	77.5	59.6
Telugu (te)	17.8	37.8	50.9	35.5	18.1	43.8	58.3	40.1	37.3	65.5	76.4	59.7

Table 2: Test set results using models trained with Hindi questions and evaluated on questions translated in 11 different languages.

hyperparameters. Table 4 shows different group sizes. We see that having group sizes that are too large or too small are detrimental to performance. Note that each group contains an average of $14n$ chunks. We envisage that, with small groups there is not much diversity in negatives and with large groups a lot of potential negatives are never used since we only take minimum number of possible negatives across chunks in a batch (as described in Section 4.3). Hence, we train with $n = 4$.

n	R@1	R@5	R@10	R-avg
2	**18.7**	43.4	59.5	40.5
4	18.0	**43.8**	**66.3**	**42.7**
8	18.4	44.5	59.9	40.9

Table 4: Tuning hyperparameter group size n, $|B| = 4, D = 2, \sigma = 2.5$. Metrics reported on Dev set using INDENT.

Data augmentation hyperparameter D creates more groups of different segment combinations. Table 5 shows how data augmentation helps, but at the cost of overfitting. $D = 3$ leads to overfitting; $D = 2$ gives the best trade-off between train and dev set performance.

D	R@1	R@5	R@10	R-avg
1	13.4	39.3	57.9	36.9
2	**18.0**	43.8	**66.3**	42.7
3	16.9	**46.1**	65.3	**42.8**

Table 5: Tuning hyperparameter data augmentation D, $|B| = 4, n = 4, \sigma = 2.5$. Metrics reported on Dev set using INDENT.

In Table 6, we present experiments with varying batch sizes. Similar to group size, we do not want the batches to be too large or too small. Although negative sampling is within a group, the number of negatives are the same within the batch and hence similar restrictions hold as with group sizes. Based on the results in Table 6, we set batch size $|B| = 4$.

$\|B\|$	R@1	R@5	R@10	R-avg
2	16.1	43.0	60.4	39.8
4	**18.0**	**43.8**	**66.3**	**42.7**
8	17.0	42.6	61.6	40.4
12	15.0	40.6	56.3	37.3

Table 6: Tuning hyperparameter batch size $|B|, n = 4, D = 2, \sigma = 2.5$. Metrics reported on the Dev set using INDENT.

In Table 7, we show experiments on tuning the standard deviation σ of the Gaussian distribution from equation 3. Using equation 2, we vary σ and scale it with α. In Table 7, we present the metrics across various α values and constant σ values. Our intuition behind using high σ in the center was that central chunks are hard to anchor, but Table 7 suggests otherwise. We see that constant σ is more effective by a significant margin, indicating that even central chunks are strongly anchored to some question.

Standard deviation		R@1	R@5	R@10	R-avg
Fixed σ	0.2	19.7	**47.1**	66.8	44.5
	0.5	**20.3**	46.9	**67.5**	**44.9**
	1	19.6	42.7	62.2	41.5
	1.5	19.6	44.1	64.7	42.8
	2.5	18.0	43.8	66.3	42.7
	3	18.1	43.5	65.2	42.3
Varying α	0.1	15.8	41.8	57.3	38.3
	0.25	15.9	42.4	58.8	39.0
	0.4	18.0	43.7	60.0	40.6
	0.6	17.9	43.6	63.0	41.5
	0.75	18.1	43.0	65.2	42.1
	0.9	17.5	40.4	63.6	40.5

Table 7: Metrics reported on Dev set with tuning σ and α; $|B| = 4, n = 4, D = 2$ and INDENT architecture is used.

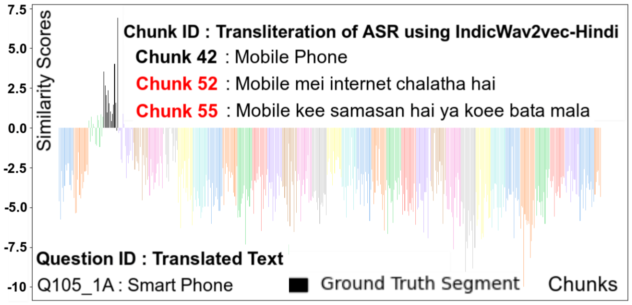

Figure 4: Chunk vs. question dot-product scores using INDENT. Same colored chunks belong to the same segment; black refers to ground-truth. Note that the question and the ASR transcript of the ground-truth segment are highly correlated. Chunks 52 and 55 have high non-negative scores due to the presence of the word *mobile* and the question asked about *smart phone* which matches exactly with chunk 42 *mobile phone*.

Analysis. We qualitatively analyze INDENT by plotting all the chunk scores for a given question. As shown in Figure 4, INDENT is able to correctly identify the ground truth segment (in black); the dot-product scores are also very high for the ground truth segment compared to the rest.

6 Conclusions and Future Work

We present a new end-to-end weakly-supervised approach INDENT, to align audio recordings with questions and an associated dataset of long health audio surveys collected from young mothers residing in rural areas of Bihar (India). INDENT automatically grounds questions within long audio recordings with the use of a Gaussian-weighted cross-attention mechanism that exploits the fact that questions appear temporally ordered in the training audio segments. We show through extensive experiments that we are able to isolate questions within long audio recordings reasonably well and also demonstrate how this framework can be used with queries in other Indian languages.

Acknowledgments

Unlinked anonymized data used for the work reported here was collected by CARE India Solutions for Sustainable Development, as part of system strengthening and governmental program implementation tracking and supported by Bill and Melinda Gates Foundation. The technical and legal support was provided by the Koita Centre for Digital Health (KCDH), and specifically by Dr Raghavendran Lakshmi Narayanan, the Health Information Manager at KCDH. Shubham Nemani and Pranjal Saini's invaluable contributions and assistance during the initial data processing stages are deeply appreciated.

References

[Anguera *et al.*, 2014] Xavier Anguera, Jordi Luque, and Ciro Gracia. Audio-to-text alignment for speech recognition with very limited resources. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[Anne Hendricks *et al.*, 2017] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pages 5803–5812, 2017.

[Ao *et al.*, 2021] Junyi Ao, Rui Wang, Long Zhou, Chengyi Wang, Shuo Ren, Yu Wu, Shujie Liu, Tom Ko, Qing Li, Yu Zhang, et al. Speecht5: Unified-modal encoder-decoder pre-training for spoken language processing. *arXiv preprint arXiv:2110.07205*, 2021.

[Baevski *et al.*, 2020] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

[Bredin *et al.*, 2020] Hervé Bredin, Ruiqing Yin, Juan Manuel Coria, Gregory Gelly, Pavel Korshunov, Marvin Lavechin, Diego Fustes, Hadrien Titeux, Wassim Bouaziz, and Marie-Philippe Gill. Pyannote. audio: neural building blocks for speaker diarization. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7124–7128. IEEE, 2020.

[CARE, 2023] CARE. Care india website. https://www.careindia.org/, 2023. Accessed: 2023-03-02.

[Chadha *et al.*, 2022] Harveen Singh Chadha, Anirudh Gupta, Priyanshi Shah, Neeraj Chhimwal, Ankur Dhuriya, Rishabh Gaur, and Vivek Raghavan. Vakyansh: Asr toolkit for low resource indic languages. *arXiv preprint arXiv:2203.16512*, 2022.

[Chen *et al.*, 2018] Yi-Chen Chen, Chia-Hao Shen, Sung-Feng Huang, and Hung-yi Lee. Towards unsupervised automatic speech recognition trained by unaligned speech and text only. *arXiv preprint arXiv:1803.10952*, 2018.

[Chen *et al.*, 2020] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[Chen *et al.*, 2022] Zhehuai Chen, Yu Zhang, Andrew Rosenberg, Bhuvana Ramabhadran, Pedro Moreno, Ankur Bapna, and Heiga Zen. Maestro: Matched speech text representations through modality matching. *arXiv preprint arXiv:2204.03409*, 2022.

[Chung *et al.*, 2018] Yu-An Chung, Wei-Hung Weng, Schrasing Tong, and James Glass. Unsupervised cross-modal alignment of speech and text embedding spaces. *Advances in neural information processing systems*, 31, 2018.

[Conneau *et al.*, 2020] Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. Unsupervised cross-lingual representation learning for speech recognition. *arXiv preprint arXiv:2006.13979*, 2020.

[Feng *et al.*, 2022] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[Gao *et al.*, 2017] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, pages 5267–5275, 2017.

[Gupta *et al.*, 2021] Anirudh Gupta, Harveen Singh Chadha, Priyanshi Shah, Neeraj Chhimwal, Ankur Dhuriya, Rishabh Gaur, and Vivek Raghavan. Clsril-23: cross lingual speech representations for indic languages. *arXiv preprint arXiv:2107.07402*, 2021.

[Heinritz *et al.*, 2022] Florian Heinritz, Gisela Will, and Raffaela Gentile. Surveying illiterate individuals: Are audio files in computer-assisted self-interviews a useful supportive tool? *Migration Research in a Digitized World*, page 101, 2022.

[Hendrycks and Gimpel, 2016] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[Javed *et al.*, 2022] Tahir Javed, Sumanth Doddapaneni, Abhigyan Raman, Kaushal Santosh Bhogale, Gowtham Ramesh, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M Khapra. Towards building asr systems for the next billion users. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10813–10821, 2022.

[Jones, 2003] Rachel Jones. Survey data collection using audio computer assisted self-interview. *Western journal of nursing research*, 25:349–58, 05 2003.

[Liu *et al.*, 2018] Meng Liu, Xiang Wang, Liqiang Nie, Qi Tian, Baoquan Chen, and Tat-Seng Chua. Cross-modal moment localization in videos. In *Proceedings of the 26th*

ACM international conference on Multimedia, pages 843–851, 2018.

[López and Luque, 2022] Fernando López and Jordi Luque. Iterative pseudo-forced alignment by acoustic ctc loss for self-supervised asr domain adaptation. *arXiv preprint arXiv:2210.15226*, 2022.

[Mithun *et al.*, 2019] Niluthpol Chowdhury Mithun, Sujoy Paul, and Amit K Roy-Chowdhury. Weakly supervised video moment retrieval from text queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11592–11601, 2019.

[Ramesh *et al.*, 2022] Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages. *Transactions of the Association for Computational Linguistics*, 10:145–162, 02 2022.

[Ravanelli *et al.*, 2021] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, Renato De Mori, and Yoshua Bengio. SpeechBrain: A general-purpose speech toolkit, 2021. arXiv:2106.04624.

[Reed *et al.*, 2014] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

[Reichmann *et al.*, 2010] William M Reichmann, Elena Losina, George R Seage III, Christian Arbelaez, Steven A Safren, Jeffrey N Katz, Adam Hetland, and Rochelle P Walensky. Does modality of survey administration impact data quality: audio computer assisted self interview (acasi) versus self-administered pen and paper? *PloS one*, 5(1):e8728, 2010.

[Synnaeve *et al.*, 2019] Gabriel Synnaeve, Qiantong Xu, Jacob Kahn, Tatiana Likhomanenko, Edouard Grave, Vineel Pratap, Anuroop Sriram, Vitaliy Liptchinsky, and Ronan Collobert. End-to-end asr: from supervised to semi-supervised learning with modern architectures. *arXiv preprint arXiv:1911.08460*, 2019.

[Wiseman, 2017] John Wiseman. Python WebRTC-VAD website. https://pypi.org/project/webrtcvad/, 2017. Accessed: 2023-06-13.

[Xu *et al.*, 2018] Huijuan Xu, Kun He, Leonid Sigal, Stan Sclaroff, and Kate Saenko. Text-to-clip video retrieval with early fusion and re-captioning. *arXiv preprint arXiv:1804.05113*, 2(6):7, 2018.

[Zhang *et al.*, 2022] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. The elements of temporal sentence grounding in videos: A survey and future directions. *arXiv preprint arXiv:2201.08071*, 2022.