HarperValleyBank: A Domain-Specific Spoken Dialog Corpus

Mike Wu[1], **Jonathan Nafziger**[2], **Anthony Scodary**[2], **Andrew Maas**[1]
Stanford University[1], Gridspace, Inc.[2]
{wumike,amaas}@cs.stanford.edu, {jonathan,anthony}@gridspace.com

Abstract

We introduce HARPERVALLEYBANK, a free, public domain spoken dialog corpus. The data simulate simple consumer banking interactions, containing about 23 hours of audio from 1,446 human-human conversations between 59 unique speakers. We selected intents and utterance templates to allow realistic variation while controlling overall task complexity and limiting vocabulary size to about 700 unique words. We provide audio data along with transcripts and annotations for speaker ID, caller intent, dialog actions, and emotional valence. The size and domain specificity of this data makes for quick experiments with modern end-to-end neural approaches. Further, we provide baselines for representation learning and transfer tasks. These experiments adapt recent work to embed utterances and use the resulting representations in prediction tasks. Our experiments show that tasks using our annotations are sensitive to both the model choice and corpus size for representation learning approaches.

1 Introduction

Recent innovations in deep learning approaches substantially improved spoken dialog systems in both academic research and industry applications. Speech recognition systems now regularly leverage neural network acoustic models to achieve near human performance (Hinton et al., 2012; Saon et al., 2017; Xiong et al., 2018). Modern systems increasingly use end-to-end recurrent neural network approaches which encode few assumptions and rapidly adapt to new data (Hannun et al., 2014; Chan et al., 2016; Maas et al., 2015; Likhomanenko et al., 2019; Graves and Jaitly, 2014; Bahdanau et al., 2016). In parallel, approaches to spoken and text-based dialog systems increasingly leverage neural networks for dialog management and state representation (Khatri et al., 2018; Andreas et al., 2016; Liu and Lane, 2017). Building and experimenting with neural network approaches often requires sufficiently large datasets and significant computational resources for training and evaluation.

Most deep learning systems require dense annotations that scale with dataset size, posing a challenging barrier to widespread adoption. In response, there has been significant recent work in representation learning for domain and task transfer via embedding models (Oord et al., 2018; Chorowski et al., 2019; Schneider et al., 2019; Baevski et al., 2020), which can be trained without any supervision and reused for many downstream tasks like predicting speaker identity (Panayotov et al., 2015; Nagrani et al., 2017) and commands (Warden, 2018; Lugosch et al., 2019). Representation transfer is important to warm start deep learning based dialog systems on new task domains.

Recent research on representaion learning for audio often use small prototyping datasets like TIMIT (Garofolo et al., 1993) and AudioMNIST (Becker et al., 2018). There are a range of existing datasets for dialog system research including spoken and text dialogs and interactions with human agents or deployed systems. See Serban et al. (2015) for a comprehensive review of available dialog datasets.

We developed the HARPERVALLEYBANK corpus for homeworks and projects in Stanford's course *Spoken Language Processing* [1], as well as research on spoken language representation learning and transfer. The goals of the corpus are to provide:

- Freely available data for education, research, or commercial development. We release the data using a Creative Commons license (CC-BY)[2].

[1] http://cs224s.stanford.edu/
[2] creativecommons.org/licenses/by/4.0

- Sufficient size and variability to meaningfully train and evaluate end-to-end neural approaches for speech transcription.

- Manageable overall size and complexity to enable students to quickly iterate on experiments without requiring expensive compute hardware for training.

- Annotations for dialog-relevant tasks aside from speech transcription (e.g. intent, dialog action) to enable multi-task training and representation transfer benchmark tasks.

- Realistic domain-specific, goal-oriented conversations to evaluate representation transfer approaches across domains in spoken dialog systems.

We recorded two-sided phone conversations to simulate customer call center interations in a financial services domain. The dataset is representative of human to human goal-oriented dialogs for consumer banking with a narrowly scoped set of intents. To evaluate transfer to tasks in our data, we borrow inspiration from recent representation learning methods in computer vision. An encoder is learned on a large, unlabeled dataset and used to represent data for supervised objectives on new datasets (Oord et al., 2018; Wu et al., 2018; Zhuang et al., 2019; Bachman et al., 2019; Misra and Maaten, 2020; He et al., 2020; Chen et al., 2020).

In the next sections, we provide more details on the corpus and its collection, followed by experiments showcaseing its applications to automatic speech recognition and unsupervised learning. In Sec. 2, we discuss basic corpus statistics, caller intents, and the data generation and annotation process. In Sec. 3, we explore several popular end-to-end neural models with multi-task objective functions to simultaneously perform speech-to-text transcription, speaker identification, and caller intent prediction. In Sec. 4, we explore using speaker identity and caller intents as downstream objectives to evaluate representation transfer, and propose a collection of unsupervised speech baselines. The full dataset with a PyTorch implementation reproducing the speech recognition and transfer experiments is available at `https://github.com/cricketclub/gridspace-stanford-harper-valley`.

2 The HarperValleyBank Corpus

We compile a dataset of recorded audio conversations between an agent and a customer of a bank. Conversations are goal-oriented, such as ordering a new checkbook or checking the balance of their account. Fig. 1 shows an example converation from the dataset. We collected data using the Gridspace Mixer platform, where crowd workers are randomly paired for short telephone conversations. Mixer membership includes hundreds of past and current professional call center agents who are trained to perform assorted Mixer tasks in domains including healthcare, telecommunications, financial services, and commerce.

2.1 Data Collection Procedure

Using the Mixer web platform, a person is randomly assigned the role of agent or customer and provided a script for the interaction along with a telephone number to call to start the conversation. Roles are randomly assigned for each call, so the same worker can appear as both customer and agent in different conversations. We created a set of conversation goals and scripts for each interaction using templates intended to capture variety in each intent while keeping workers' word choices and the overall interactions fairly simple with limited vocabulary. We do not control the noise environment or microphones used by each worker, and there is natural variation across different types of phones and environments.

When a person calls in, they are placed on hold until they are paired with the next available conversation partner. The groups are large enough that many unique pairings occur over the course of one session. Once the Mixer task is live for the caller, the web application will change state, informing the caller whether they are acting in the role of the agent or caller. The instructions, data, and user interface adapt to the role and provide a rough script. The customer role initiates a call task by expressing an intent, and the agent role has an interactive web interface to simulate completing a task. We encouraged callers to use mobile phones or headsets to encourage a microphone transfer function that is acoustically representative of a real call center.

During each call, participants have the script for their side of the conversation in front of them in a web browser. We provide examples of suggested phrasings in the Supplemental Material. The worker playing the customer role is given a single intent for the conversation, along with specific values for relevant slots (e.g. the amount of money to transfer and the source/target accounts). When playing the agent role, a worker is shown some simple buttons and menus they must click to perform the requested operation (e.g. ”check account balance”, or ”transfer money”).

A conversation is deemed successful and considered for the dataset if the agent correctly executes the task provided to the customer caller. Names and slot values for different transactions are randomly generated, and we limit the number of possible names and proper nouns to reduce overall corpus vocabulary size. The Gridspace Mixer platform handles generating random templates from a high level specification, all associated telephony operations to pair callers, and recording audio along with metadata for each interaction.

AGENT: hello this is harper valley national bank my name is jay how can I help you today
CALLER: hi my name is mary davis
CALLER: [noise]
CALLER: i would like to schedule an appointment
AGENT: yeah sure what day what time
CALLER: thursday one thirty p m
AGENT: that's done anything else
CALLER: that's it
AGENT: have a good one.

Figure 1: Example Conversation

2.2 Data Labelling

Gridspace Mixer trains a subset of its community to perform a wide range of annotations. For this corpus, Mixers performed three primary labeling tasks: *text transcript, audio quality, and script adherence ratings*. Gridspace has provided the Mixer community with a highly specialized speech labeling tool called Scriber. Scriber is designed for rapid human transcription and data labeling. The tool also provides a wide array of convenience and ergonomic functions, designed to enable efficient labeling of large spoken language datasets. Every person trained to use Scriber must go through several training sessions, which requires them to watch training videos and perform well on a quiz. For dialog actions and emotional valence, labels were instead produced using a Gridspace API rather than human annotation. As a result, there may be some noise or bias, but our experiments indicate they are reasonable for a benchmark task.

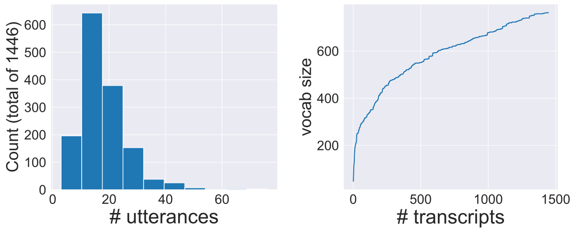
The HARPERVALLEYBANK corpus was collected over three separate Mixer sessions and then filtered post-annotation, informed by the script adherence labels and audio quality labels, to ensure the data was simple and low variance. This filtering ensures the corpus provides conversational and task-oriented speech data while regulating for simplicity. The primary target of the cleaning were conversations where calls dropped or there were other technical issues which derailed the conversation. Specifically we removed conversations with script adherence ratings less than 4 and audio quality ratings less than 3. Furthermore we filtered out conversations which contained some words such as 'frozen', 'website', and 'refresh', which indicated conversation about technical issues with the task interface. In total we removed 375 conversations.

2.3 Corpus Statistics

HARPERVALLEYBANK Statistics	
Hours of audio	23.7
# of conversations/transcripts	1,446
# of utterances	25,730
# of unique words	735
Mean # of lines per conversation	17.8
Median # of lines per conversation	16
Mean # of words per utterance	4.1
Median # of words per utterance	4.5
# of unique speakers	59
# of task classes	8
# of dialog action classes	16
# of sentiment classes	3

Table 1: Basic statistics of the corpus.

Table 2 shows basic statistics of the HARPERVALLEYBANK corpus. The corpus contains about 23 hours of audio in total, across 1,446 conversations. Conversations range from 2 to 60 utterances, with an average of 18. Each utterance roughly corresponds to a single turn in the conversation. Due to automatic segmentation of utterances, there can be multiple utterances in a row from a single speaker's turn. Notably, the corpus has a small vocabulary of approximately 700 unique words. Many of the most common words in the vocabulary are domain-specific to customer service e.g. “help”, “thank”, or “please”. Fig. 2b depicts how vocabulary size



(a) (b)

Figure 2: (a) The empirical distribution over the number of utterances in a conversation transcript. (b) The size of the vocabulary, or number of unique words spoken, as a function of the number of conversations.

scales with dataset size.

The Gridspace platform records each side of the conversation separately, and we release the audio in speaker-separated files encoded as 8kHz per the original telephony data. We transcribed the utterances via crowd workers with basic speech transcription training again using the Gridspace Mixer platform. Workers are not instructed to carefully transcribe word fragments or non-speech noises. Leveraging crowd workers and transcribing without precise fragments and non-speech tags has been shown to be a viable approach for training speech recognition systems (Novotney and Callison-Burch, 2010). We also provide estimated alignments using the Gridspace Speech Recognition API, although we did not evaluate the accuracy of these automatic alignments.

In addition to human transcriptions of each conversation, we include four categories of metadata to create additional inference tasks:

- **Intent**. Each conversation has a single intent representing the customer's goal in the conversation. An intent can be one of eight categories: *order checks, check balance, replace card, reset password, get branch hours, pay bill, schedule appointment, transfer money*. The distribution over intents is roughly balanced, and derived automatically from the tasks assigned to callers during collection. Fig 3a shows the distribution of intent labels for conversations.

- **Emotional Valence**. Utterances are automatically labeled with three sentiment categories, *negative, neutral, and positive*. There is a probability estimate label for each category. These annotations are generated by a model in the Gridspace Speech API which was trained on a large corpus of proprietary data from multiple domains. Fig 3b shows the distribution of probabilities for each sentiment category across utterances.

- **Speaker ID**. Utterances have a unique ID out of 59 speakers. The number of utterances per speaker are imbalanced, with most speakers responsible for less than 50 utterances. Fig 3c shows the distribution of utterances by speaker.

- **Dialog Action**. A label per utterance corresponding to types of "conversational move." represented in the utterance. There are 16 total dialog actions, and more than one can be present in an utterance. Like speaker identity, the distribution over actions is imbalanced with "greeting" being the most frequent and many infrequent actions combined into the "other" category. The 16 possible actions are: *"yes" response, greeting, response, data confirmation, procedure explanation, data question, closing, data communication, "bear with me" response, acknowledgement, data response, filler disfluency, thanks, open question, problem description*, and *other*. Fig 3d shows the distribution of dialog actions for utterances.

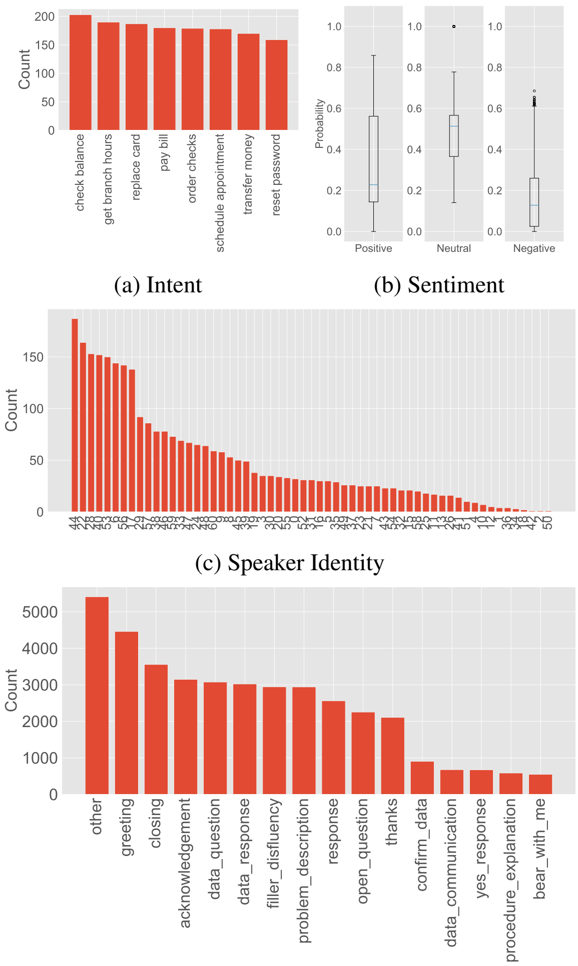
3 Spoken Language Understanding

We now consider some baselines for the HARPERVALLEYBANK corpus on speech recognition and our four prediction tasks. For speech recognition, we focus on three common approaches: connectionist temporal classification or CTC (Graves et al., 2006), Listen-Attend-Spell or LAS (Chan et al., 2016), and finally, a "multi-task" objective combining the two previous losses (Kim et al., 2017), MTL. All objectives produce an *embedding* vector by encoding audio features.

In addition to optimizing the speech recognition objective, denoted $\mathcal{L}_{\text{asr}}$, we fit four linear layers mapping the embedding of the audio signal to a prediction for each of the other tasks: the speaker identity, intent, dialog action, and sentiment labels. These four auxiliary objectives are optimized jointly with the speech recognition objective:

$$\mathcal{L}_{\text{asr}} + \mathcal{L}_{\text{spk}} + \mathcal{L}_{\text{task}} + \mathcal{L}_{\text{action}} + \mathcal{L}_{\text{sent}} \quad (1)$$

where $\mathcal{L}_{\text{spk}}$, $\mathcal{L}_{\text{task}}$, and $\mathcal{L}_{\text{sent}}$ are cross entropy losses, whereas $\mathcal{L}_{\text{action}}$ comprises a sum of binary



(a) Intent (b) Sentiment

(c) Speaker Identity

(d) Dialog Action

Figure 3: Distribution over auxiliary labels in the dataset. Subfigures (a,c,d) show the counts for intent, speaker, and dialog action, respectively. Subfigure (b) show boxplots for each of the three sentiments.

cross entropy losses, one for each of the sixteen dialog actions, of which more than one may apply to any single utterance.

Training Details For the ASR models, audio waveforms are loaded by `torchaudio` and preprocessed to log Mel-freqency spectrogram features using Librosa (McFee et al., 2017). For CTC, we encode log-Mel features using a bi-directional LSTM with two layers and 128 hidden dimensions. CTC decoding is done greedily with no language model. In LAS, the listener network is composed of three stacked pyramid bi-directional LSTMs (which cuts the input length by 4) with 128 hidden dimensions whereas the speller network is an uni-directional LSTM with two layers, 256 hidden dimensions, and a single-head attention layer. For MTL, we use the same encoder as LAS but include both the speller and CTC decoder. As encoder parameters are shared, we can interpret CTC as a secondary objective whose main role is to regularize the LAS encoder to respect CTC alignments. The LAS and MTL objectives both use a label smoothing hyperparameter of 0.1 and teacher forcing with probability 0.9. We train each model for 200 epochs with Adam (Kingma and Ba, 2014) using a learning rate of 5e-4, weight decay 1e-5, batch size 256 with gradient clipping.

Results and Analysis Table 2 shows the performance of CTC, LAS, and MTL. As consistent with prior literature, we find LAS to outperform CTC in terms of word error rate (WER) since the latter is forced to assumed conditional independence among input characters. However, we surprisingly find CTC to outperform LAS in all four auxiliary tasks, suggesting that the learned alignments enforced by CTC may be useful for speaker ID and intent prediction, among others. Further, we find MTL to outperform both LAS and CTC in word error rate, reproducing the findings from (Kim et al., 2017), with equivalent performance as LAS on the auxiliary tasks. In contrast, Table 3 shows results on a speaker-disjoint split of the data with speaker ID results omitted by necessity. Interestingly, we no longer see CTC outperforming the end-to-end neural approaches on auxiliary tasks, suggesting that CTC has lacks generalization to unobserved speakers. Once again, we find LAS and MTL to greatly outperform CTC in WER.

4 Representation Learning

Unsupervised representation learning seeks to derive useful representations of unstructured data, such as speech waveforms, without any human annotations. (Unlike the experiments above, we no longer assume access to human transcriptions or classification labels of identity or action.) The learned representations are considered a "summary" of the raw data, often being used as a starting point for downstream tasks. For instance, unsupervised speech representations might be used to predict speaker identity or utterance topic. As HARPERVALLEYBANK is a fairly small dataset, it is a suitable candidate dataset to measure the effectiveness of pretrained speech representations.

To showcase a strong family of baselines, we borrow recent ideas from contrastive learning (Wu et al., 2018; Zhuang et al., 2019; Bachman et al., 2019; Misra and Maaten, 2020; He et al., 2020;

Model	WER	Speaker ID	Intent	Dialog Action (F1)	Sentiment
CTC	46.9	**89.5**	**47.8**	**37.4**	**82.4**
LAS	13.3	88.0	30.8	26.8	72.1
MTL	**12.7**	88.3	30.7	27.9	72.4

Table 2: Performance on the HARPERVALLEYBANK test set. Word error rate (WER) of speech recognition systems along with accuracy on auxiliary dialog tasks: speaker ID, caller intent, dialog action, and sentiment predition. For dialog action, we report F1 scores. Examples are split by utterance randomly.

Model	WER	Intent	Action (F1)	Sentiment
CTC	51.6	36.7	32.9	75.3
LAS	12.7	36.9	**34.5**	**80.0**
MTL	**11.5**	**38.2**	33.7	76.1

Table 3: Speech recognition and auxiliary task performance on HARPERVALLEYBANK *split by speaker*.

Chen et al., 2020; Wu et al., 2020; Joshi et al., 2020) where the goal is to learn good representations by discriminating between specific instances in a dataset. In particular, we adapt four algorithms from computer vision: Instance Discrimination or IR (Wu et al., 2018), Local Aggregation or LA (Zhuang et al., 2019), Momentum Contrast or MoCo (He et al., 2020), and SimCLR (Chen et al., 2020). We provide a brief summary of each algorithm with necessary edits for usage in speech.

4.1 Contrastive Learning for Audio

Let $\mathcal{D} = \{x_i\}_{i=1}^n$ be a dataset of n speech waveforms sampled independently from a distribution $p(x)$, and let $\mathcal{T}$ be family of data augmentations such that every $t : X \to X \in \mathcal{T}$ is a function mapping one waveform to another. For instance, adding noise to or cropping are two common augmentations of speech waveforms. Suppose we have a distribution $p(t)$ over the functions in $\mathcal{T}$, often chosen to be the uniform distribution. Now, introduce an *encoder*, a neural network that maps x to a representation $g_\theta(x)$, which is L_2 normalized to prevent trivial solutions. The contrastive objective for the i-th example is:

$$\mathcal{L}(x_i) = \log \frac{e^{g_\theta(t(x_i))^T g_\theta(t'(x_i))/\tau}}{\sum_{j\in\{i,1:k\}} e^{g_\theta(t(x_i))^T g_\theta(t_j(x_j))/\tau}} \tag{2}$$

where $t, t', t_{2:k} \sim p(t)$ and $x_{1:k} \sim p(x)$ i.i.d. (We assume this to be the case from now on unless otherwise mentioned.) We call $x_{1:k}$ negative samples: if we consider the numerator of Eq. 2 as a similarity function between two terms, the denominator seeks to normalize that similarity with respect to other plausible examples drawn from the dataset.

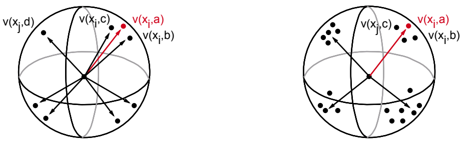
Prior work has shown that Eq. 2 is a lower bound on the mutual information between two views of an instance (Bachman et al., 2019; Chen et al., 2020; Tian et al., 2020; Wu et al., 2020). Optimizing this would be trivial if not for the augmentation functions, which we often choose to hide information in x. For instance, adding white noise or cropping an audio waveform bottlenecks information. By maximizing mutual information despite these lossy transformations, the representations are encouraged to be abstract and invariant. In this work, we construct $\mathcal{T}$ for by (1) randomly taking contiguous subsets of the raw wavform (i.e. cropping), and (2) adding white noise to the raw waveform. In our experiments, we will also explore masking time and spatial frequenties in the log-Mel spectrograms, rather than applying augmentations at the waveform level. The choice of augmentations is fundamental to the success of contrastive algorithms, much more than the typical role of data augmentation in supervised training. Future work should explore "optimal" choices for audio views, focusing on domain specific transformation e.g. pitch, speed, or adding background noise.

In practice, the number of negative examples, k, needs to be large to reduce variance in its estimate of the partition function. As computing the denominator requires $k+1$ forward (and backward) passes through the encoder, this becomes quickly intractable. No shortage of technical innovation has gone to circumventing this expensive operation. We focus on four different approaches.

IR The innovation of IR is to introduce a memory bank M that stores the embedding for the i-th entry in the training dataset throughout training. That is, given the current minibatch containing the i-th example x_i, we compute $g_\theta(t(x_i))$ using a random view of x_i and save it to entry $M[i]$. Having done so, we can rewrite Eq. 2 as:

$$\mathcal{L}_{\mathrm{IR}}(x_i) = \log \frac{e^{g_\theta(t(x_i))^T M[i]/\tau}}{\sum_{j\in\{i,j_{1:k}\}} e^{g_\theta(t(x_1))^T M[j]/\tau}} \tag{3}$$

for the i-th example in $\mathcal{D}$, where $j_{1:k}$ are uniformly chosen from $[1, n]$. Observe that this is equivalent to choosing negatives from $p(x)$ as each row in M corresponds to an instance in $\mathcal{D}$. At the current minibatch, $M[i]$ stores the representation of the augmentation of i-th example from the last epoch. Thus, the numerator of Eq. 3 still compares two augmentations of x_i. The benefit of IR is that retrieving from $M[i]$ is almost no cost, meaning we are free to choose k to be very large. The disadvantage however, is that gradients cannot propagate through M and the entries in M can be stale.



(a) IR (b) LA

Figure 4: Comparison of "optima" for IR and LA. For intuition, the optima of contrastive learning is to spread the embeddings of data points uniformly on the surface of a d dimensional sphere, where d is embedding dimensionality. Such a configuration makes it such that any pair of points is as easy to discriminate as any other pair, a property that may be useful for transfer learning.

MoCo To combat staleness, MoCo replaces the memory bank with a first-in-first-out (FIFO) queue of size k. Every minibatch, the latest representations are cached into the queue while the most stale ones are removed. Additionally, MoCo introduces a second (momentum) encoder $g'_{\theta'}$. Now, the primary encoder g_θ is used to embed one view of the instance x_i whereas the momentum encoder is used to embed the other. Again, gradients are not propagated to $g'_{\theta'}$. Instead, its parameters are deterministically set by a momentum equation $\theta' = m\theta' + (1 - m)\theta$ where m is an update coefficient. The MoCo objective is

$$\mathcal{L}_{\text{MoCo}}(x_i) = \log \frac{e^{g_\theta(t(x_i))^T g'_{\theta'}(t'(x_i))/\tau}}{\sum_{j \in \{1:k\}} e^{g_\theta(t(x_i))^T Q[j]/\tau}} \quad (4)$$

where Q is the FIFO queue.

LA Building on IR, the LA algorithm treats neighbor examples of x_i as its augmentations. For motivation, two customers calling the bank to order a checkbook might be good augumentations of each other: the audio may differ in word choice, but the semantic meaning is identical. Doing so may encourage the representation to be invariant to syntax. Intuition-wise, using neighboring examples as augmentations is equivalent to encouraging uniform clusters of points on the hypersphere, as opposed to uniform points (see Fig. 4).

More precisely, LA introduces two new sets for every input x_i: a *background* neighbor set B_i and a *close* neighbor set C_i, each containing indices between 1 and n representing the elements in $\mathcal{D}$ belonging to each set. For x_i, the background neighbor set contain examples in $\mathcal{D}$ most similar to $M[i]$ as measured by dot product in embedding space. The close neighbor set is defined separately by elements that belong to the same cluster as $M[i]$ using an ensemble of K-Means clusterings optimized using all embeddings stored in M. We assume $C_i \subseteq B_i$. Now, the LA loss is:

$$\mathcal{L}_{\text{LA}}(x_i) = \log \frac{\sum_{k \in C_i} e^{g_\theta(t(x_i))^T M[k]/\tau}}{\sum_{k' \in B_i} e^{g_\theta(t(x_i))^T M[k']/\tau}} \quad (5)$$

In vision, LA outperforms IR in transfer tasks. We study if the benefits generalize to speech.

SimCLR Finally, we discuss a more recent contrastive algorithm that outperforms IR, MoCo, and LA, nearing supervised performance in vision. One of the fundamental flaws of the previous three algorithms is being forced to cut gradients to half of the terms in the objective. From an optimization perspective, this should dramatically reduce the amount of learning signal in each gradient step. The SimCLR framework (Chen et al., 2020) circumvents this by bootstrapping negative samples from the other elements in same minibatch as the current example. Since members of a minibatch are sampled uniformly from the dataset, doing so is unbiased. Further, since neural networks do computation one minibatch at a time, this procedure introduces little additional compute. Precisely, suppose every iteration we are given a minibatch of waveforms with two views each, denoted $x_{1:m}$ and $x_{m:2m}$ such that x_i and x_{i+m} are views of the same image, m being the minibatch size.

In summary, the SimCLR objective is

$$\mathcal{L}_{\text{SimCLR}}(x_i) = \log \frac{e^{g_\theta(t(x_i))^T g_\theta(t'(x_{i+m}))/\tau}}{\sum_{j \neq i, j=1}^{2m} e^{g_\theta(t(x_i))^T g_\theta(t_j(x_j))/\tau}} \quad (6)$$

where $i \in [1, m]$ and $x_{1:2m} \sim p(x)$ i.i.d. The efficacy of SimCLR highly depends on m being large, otherwise reverting to the issue of high variance from using too few negative samples. We are again interested to see if the benefits of

SimCLR generalize to the audio domain.

Since these methods were primarily developed for vision, they have not been extensively applied to speech. One of the contributions of this paper is to establish comparable baselines for audio representation learning across this suite of recent algorithms. As a close relative, we also compare to the representations learned using Wav2Vec-1.0 (Schneider et al., 2019) and Wav2Vec-2.0 (Baevski et al., 2020), contrastive algorithms that are trained to predict future audio sequences, instead of another view of the same audio sequence.

As these unsupervised representations are posited to be general, we can measure their usefulness on a wide variety of transfer tasks by fitting a downstream model on top of the pretrained, and frozen, embeddings. A "better" representation should result in a higher classification accuracy across the four HarperValleyBank caller intents, which are diverse enough to gauge different properties of the representation. We purposefully fit a small linear model, e.g. logistic regression, (and no finetuning) for each task separately to focus on analyzing the embedding quality.

Model	Spk.	Intent	Action	Sent.
Wav2Vec 1.0 (960hr)	18.2	17.1	0.0	53.7
Wav2Vec 2.0 (100hr)	22.3	19.7	0.0	54.3
Wav2Vec 2.0 (960hr)	27.3	20.5	0.0	55.5
IR (100hr)	99.5	99.1	0.0	51.3
LA (100hr)	99.5	98.8	0.0	50.5
MoCo (100hr)	99.6	98.9	0.0	53.2
SimCLR (100hr)	99.8	99.3	0.0	53.9
IR* (100hr)	99.5	84.5	0.0	51.4
LA* (100hr)	97.5	75.8	1.4	55.1
MoCo* (100hr)	99.1	82.6	0.0	54.0
SimCLR* (100hr)	99.2	81.4	0.0	54.6
IR (960hr)	99.9	99.9	17.4	66.3
LA (960hr)	99.9	99.9	18.4	64.6
MoCo (960hr)	99.9	99.9	17.3	65.5
SimCLR (960hr)	99.9	99.9	17.4	65.9
IR* (960hr)	99.9	86.7	17.6	64.8
LA* (960hr)	99.9	79.8	18.0	64.6
MoCo* (960hr)	99.5	86.1	16.2	64.3
SimCLR* (960hr)	98.6	82.6	16.1	65.6

Table 4: Performance on speaker identity, intent, dialog action, and sentiment. We report F1 score for dialog action. The superscript (*) represents using spectral augmentations rather than wavform augmentations.

Training Details We again ignore speakers with less than 10 utterances in the dataset. For Wav2Vec based models, we use the official implementation and pretrained weights available on FairSeq (Ott et al., 2019). All algorithms are trained on either the 100 hour split or 960 hour split of LibriSpeech (Panayotov et al., 2015). To fit IR, LA, MoCo, and SimCLR, we first apply any data augmentations, then truncate to waveforms to 150k frames, and compute the log-Mel spectrogram as the input to the encoder. Spectrograms are z-scored using the mean and standard deviation computed from the training split, which we found to be important for generalization to new domains. By default, our augmentations select contiguous crops of waveforms with a minimum and maximum ratio of 0.08 to 1.0, along with Gaussian noise with a scale of 1.0. We separately explore first computing the spectrogram, then applying a time and frequency mask (using the `nlpaug` library with a mask factor of 40 for both), denoted by the (*) superscript in Table 4. Regardless, we use a hop length of 1344 and a FFT window of 112 for an processed spectrogram shape of 112 by 112. Then, we use a ResNet50 (He et al., 2016) to map this to an d=2048 dimensional embedding. Wav2Vec-1.0 and Wav2Vec-2.0 use customized architectures that amount to around 1.5 times the number of parameters as ResNet50.

After representation learning, we measure the quality of an embedding by linear classification (Wu et al., 2018; Zhuang et al., 2019; He et al., 2020; Chen et al., 2020). We use the features after the last convolutional layer and prior to the global pooling, resulting in a 2048x4x4 dimensional vector and fit a logistic regression model mapping this vector to a probability for each class in the transfer task. A separate regression is fit for speaker ID, intent, dialog action, and sentiment. For the Wav2Vec family, we use the encoded embedding, averaged over timesteps, with dimension 512 and 1024 for the 1.0 and 2.0 models, respectively. Each transfer dataset is split into train (80%) and test (20%) sets by class to ensure both sets have instances of each class. Thus, each transfer task has its own train test split (which is notably different than the one used in the ASR experiments). The same data augmentations used in pretraining are used in transfer but not in evaluation.

In optimization, we use SGD with batch size 256, learning rate 0.03, momentum 0.9, weight decay 1e-4 for 200 epochs. In transfer, we use SGD with batch size 256, learning rate 0.01, momentum 0.9, weight decay 1e-5 for 100 epochs. Due to the size of Wav2Vec models, we use a batch size of 64. In the contrastive objectives, we use a temperature τ=0.07. For IR and LA, we use 4096 negative samples and set the memory bank update parameter to 0.5. For LA, we fit KMeans with k=5000 ten times with different random seeds and take the union of all of 10 clusters with the current input as a member. To do tractable large-scale similarity search, we use the FAISS toolkit (Johnson et al., 2019). For MoCo, we use a queue of size 66536 and a momentum update of 0.99. We leave more careful hyperparameter search to future work.

Results and Analysis Table 4 reports test accuracies comparing the different unsupervised models. For dialog action prediction, we compute F1 score, which more reliably measures model performance given a biased label set (e.g. a F1 score of 0.0 corresponds to an accuracy of $\sim$90%).

We make a few observations:

- IR, LA, MoCo, and SimCLR surpass purely supervised methods (e.g. CTC, LAS, and MTL) in terms of speaker and intent prediction, although falling short in inferring dialog action and sentiment.
- The visual contrastive objectives (IR, LA, MoCo, and SimCLR) outperform the Wav2Vec family significantly on Speaker ID and caller intent with gains of 70%.
- Dialog action prediction is a surprisingly difficult task for speech representation learning. Whereas supervised methods (fit on HARPERVALLEYBANK) achieve upwards of 30.0 F1, the best models in Table. 4 achieve half the score, despite seeing 960 hours of data. Further, models trained on only 100 hours (and all Wav2Vec algorithms), do no better than trivially predicting one label.
- On the other hand, unsupervised methods reached near ceiling for caller intent prediction whereas CTC, LAS, and MTL at best, approached 40% accuracy.
- We see consistent gains in IR, LA, MoCo, and SimCLR when pretraining on 860 additional hours of data. This difference is most apparent in dialog action and sentiment prediction.
- Unlike in vision, LA, MoCo, SimCLR do not show consistent improvements over IR, suggesting that recent innovations might be overfitting to the visual modality.

In summary, we proposed three baseline algorithms for representation learning in audio with the caller intents from HARPERVALLEYBANK as measures of the usefulness of a representation. We find improvements over previous algorithms (e.g. Wav2Vec) which future research can build upon.

5 Conclusion

We introduced HARPERVALLEYBANK, a new speech corpus of transcribed conversations between employees and customers in a bank transcaction. The corpus includes additional labels, including speaker identity, caller intent, dialog actions, and sentiment. In our experiments, we established baseline models that showed this corpus to be an interesting challenge for future algorithms, and a useful educational tool for modern deep learning approaches to spoken dialog. Our experiments analyzed utterances independently, future work can explore using the HARPERVALLEYBANK corpus in conversation modelling and its related downstream optimization.

Acknowledgments

We thank the Gridspace team for their support in data collection and analysis. Experiments used PyTorch (Paszke et al., 2019), PyTorch Lightning (Falcon et al., 2019), and Weights & Biases (Biewald, 2020). We thank Dan Jurafsky for helpful discussion and feedback.

References

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Learning to compose neural networks for question answering. *arXiv preprint arXiv:1601.01705*.

Philip Bachman, R Devon Hjelm, and William Buchwalter. 2019. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15535–15545.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*.

Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. 2016. End-to-end attention-based large vocabulary speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4945–4949. IEEE.

Sören Becker, Marcel Ackermann, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. 2018. Interpreting and explaining deep neural networks for classification of audio signals. *CoRR*, abs/1807.03418.

Lukas Biewald. 2020. Experiment tracking with weights and biases. *URL https://www. wandb. com/. Software available from wandb. com*.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964. IEEE.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*.

Jan Chorowski, Ron J Weiss, Samy Bengio, and Aäron van den Oord. 2019. Unsupervised speech representation learning using wavenet autoencoders. *IEEE/ACM transactions on audio, speech, and language processing*, 27(12):2041–2053.

WEA Falcon et al. 2019. Pytorch lightning. *GitHub. Note: https://github. com/williamFalcon/pytorch-lightning Cited by*, 3.

John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. 1993. Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *STIN*, 93:27403.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Alex Graves and Navdeep Jaitly. 2014. Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning*, pages 1764–1772.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. 2014. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Chandra Khatri, Behnam Hedayatnia, Anu Venkatesh, Jeff Nunn, Yi Pan, Qing Liu, Han Song, Anna Gottardi, Sanjeev Kwatra, Sanju Pancholi, et al. 2018. Advancing the state of the art in open domain dialog systems through the alexa prize. *arXiv preprint arXiv:1812.10757*.

Suyoun Kim, Takaaki Hori, and Shinji Watanabe. 2017. Joint ctc-attention based end-to-end speech recognition using multi-task learning. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4835–4839. IEEE.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tatiana Likhomanenko, Gabriel Synnaeve, and Ronan Collobert. 2019. Who needs words? lexicon-free speech recognition. *arXiv preprint arXiv:1904.04479*.

Bing Liu and Ian Lane. 2017. Iterative policy learning in end-to-end trainable task-oriented neural dialog models. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 482–489. IEEE.

Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. 2019. Speech model pre-training for end-to-end spoken language understanding. *arXiv preprint arXiv:1904.03670*.

Andrew Maas, Ziang Xie, Dan Jurafsky, and Andrew Y Ng. 2015. Lexicon-free conversational speech recognition with neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 345–354.

Brian McFee, Matt McVicar, Oriol Nieto, Stefan Balke, Carl Thome, Dawen Liang, Eric Battenberg, Josh Moore, Rachel Bittner, Ryuichi Yamamoto, et al. 2017. librosa 0.5. 0.

Ishan Misra and Laurens van der Maaten. 2020. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717.

Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. 2017. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*.

Scott Novotney and Chris Callison-Burch. 2010. Cheap, fast and good enough: Automatic speech recognition with non-expert transcription. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 207–215.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210. IEEE.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037.

George Saon, Gakuto Kurata, Tom Sercu, Kartik Audhkhasi, Samuel Thomas, Dimitrios Dimitriadis, Xiaodong Cui, Bhuvana Ramabhadran, Michael Picheny, Lynn-Li Lim, et al. 2017. English conversational telephone speech recognition by humans and machines. *arXiv preprint arXiv:1703.02136*.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*.

Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. 2015. A survey of available corpora for building data-driven dialogue systems. *arXiv preprint arXiv:1512.05742*.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. 2020. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*.

Pete Warden. 2018. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*.

Mike Wu, Chengxu Zhuang, Milan Mosse, Daniel Yamins, and Noah Goodman. 2020. On mutual information in contrastive learning for visual representations. *arXiv preprint arXiv:2005.13149*.

Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance-level discrimination. *arXiv preprint arXiv:1805.01978*.

Wayne Xiong, Lingfeng Wu, Fil Alleva, Jasha Droppo, Xuedong Huang, and Andreas Stolcke. 2018. The microsoft 2017 conversational speech recognition system. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5934–5938. IEEE.

Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. 2019. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6002–6012.

6 Supplemental Material

6.1 Dataset Examples

We include 10 randomly chosen conversations from the HARPERVALLEYBANK Corpus.

CALLER: [noise]
CALLER: [noise]
AGENT: hello this is harper valley national bank my name is james how can i help you today
CALLER: hi my name is [unk] [unk] i need to check my account balance
CALLER: [noise]
CALLER: [noise]
AGENT: which account would you like to check
CALLER: [noise]
CALLER: my savings account
AGENT: your savings account balance is ninety five dollars
AGENT: is there anything else i can help you with
CALLER: [noise]
CALLER: no thank you that will be all for today
AGENT: thank you for calling have a great day
CALLER: you too thank you

Figure 5: Supplemental Example 1

AGENT: hello this is harper valley national bank my name is elizabeth how can i help you today
CALLER: hi my name is mary williams
CALLER: i would like to transfer money
CALLER: between my accounts
AGENT: okay one moment please
AGENT: uh what is the transfer amount
CALLER: the amount is one hundred and thirty two dollars
AGENT: what is the source account
CALLER: from my checking account
CALLER: to my savings account
AGENT: alright thank you very much your [unk] your transfer has gone through is there anything else i can help you with
CALLER: that's everything for today thank you very much
AGENT: well thank you for calling have a great day
CALLER: [unk] bye
AGENT: bye bye
CALLER: [noise]

Figure 6: Supplemental Example 2

CALLER: [noise]
AGENT: hello this is harper valley national bank
AGENT: my name is elizabeth how can i help you
CALLER: hi my name is john garcia i need to check my account balance
AGENT: which account would you like to check
CALLER: my savings account
AGENT: your savings account balance is one hundred twenty one dollars
AGENT: is there anything else i can help you with
CALLER: that'll be all thank you
AGENT: thank you for calling have a great day

Figure 7: Supplemental Example 3

6.2 Suggested Phrasings

For each task the Mixers were given suggested phrases to use in order to complete the given task. We have included these phrases below, broken down by speaker role and task. We also include agent instructions for starting and ending calls. Bolded text indicates text that was randomized for each conversation.

6.2.1 Agent Instructions

- Wait for the call to be connected...
- "Hello, this is Harper Valley National Bank, my name is *Mary*"
- "How can I help you today?"
- Use the scripts for each task.
- To ask the caller to repeat: "Can you repeat the [item-to-be-repeated]"
- Read additional response after completing each task
- When done: "Is there anything else I can help you with?"

AGENT: hello this is harper valley national bank my name is james how can i help you today
CALLER: hi my name is robert wilson
CALLER: i would like to reset my password
AGENT: okay what was your phone number
CALLER: my phone number is zero one zero
CALLER: seven eight three
CALLER: seven five two nine
AGENT: [unintelligible] zero [unintelligible] three seven five two nine
CALLER: yes
AGENT: your password reset link has been sent to your phone is there anything else i can help you with
CALLER: no that's it thank you
AGENT: and thank you for calling have a great day
AGENT: [noise]
CALLER: you too
AGENT: [noise]

Figure 8: Supplemental Example 4

- "Thank you for calling, have a great day!"

6.2.2 Schedule an appointment

Caller

- "I would like to schedule an appointment"
- When prompted for day: *"Tuesday"*
- When prompted for time: *"10:30 AM"*

Agent

- "What day would you like for your appointment?"
- "What time would you like for your appointment?"

6.2.3 Replace card

Caller

- "I lost my *credit card*, can you send me a new one?"
- If prompted for card type: "My *credit* card."

Agent

- "Which card would you like to replace"

AGENT: hello this is ms harper
AGENT: i'm sorry hello this is harper valley national bank my name is linda how can i help you today
CALLER: [noise]
CALLER: hi my name is jennifer miller i would like to pay a bill
AGENT: okay pay a bill i can help you with that
AGENT: and what's the name of the company
CALLER: the company is smart electric
AGENT: okay smart
AGENT: electric
CALLER: uh it's actually electic so it's e l e c t i c it's spelled a little funny
AGENT: and the address
AGENT: okay
AGENT: i've got that
AGENT: and um address
CALLER: the address is one three seven main street
CALLER: that's gonna be in forest ranch
AGENT: one
AGENT: main
AGENT: street
CALLER: and that's california
AGENT: [noise]
CALLER: three zero three four five
AGENT: i'm sorry can you repeat that zip [unk] code
CALLER: yeah three zero three four five
AGENT: mkay
AGENT: and what amount would you like to pay today
CALLER: the amount of the bill is ninety nine dollars
AGENT: okay is there anything else i can help with today
CALLER: no that'll be it thank you
AGENT: okay thank you for calling have a great day
CALLER: uh huh mhm bye
AGENT: alright bye bye

Figure 9: Supplemental Example 5

6.2.4 Transfer money

Caller

- "I would like to transfer money between my accounts"

- “From my *checking* account to my *savings* account”
- “The amount is *$97*”

Agent

- “What is the transfer amount?”
- “What is the source account?”
- “What is the destination account?”

6.2.5 Check account balance

Caller

- “I need to check my account balance”
- When prompted account type: “My *checking* account”

Agent

- “Which account would you like to check?”:

6.2.6 Pay a bill

Caller

- “I would like to pay a bill”
- “The company is *Fossil Gas*”
- “The address is *120 Main Street, Forest Ranch, California, 84732*”
- “The amount of the bill is *$102*”

Agent

- “What is the company name?”:
- “What is the company address?”:
- “What is the bill amount?”:

6.2.7 Order a new checkbook

Caller

- “I need a new checkbook”
- “My address is *460 First Street, Forest Ranch, California, 07307*”

Agent

- “What is your address?”

6.2.8 Reset password

Caller

- “I would like to reset my password”
- “My phone number is *497-522-3547*”

Agent

- “What is your phone number?”:

6.2.9 Get local branch hours

Caller

- “What are the local branch hours”

Agent

- “The branch hours are 9:30 am - 5:00 pm”

CALLER: [unintelligible] okay
AGENT: hello this is harper valley national bank my name is mary how can i help you today
CALLER: hello uh my name is elizabeth jones
AGENT: [noise]
AGENT: [noise]
AGENT: [noise]
CALLER: i'd like to pay a bill
AGENT: [noise]
AGENT: [noise]
AGENT: and what is the company name
AGENT: [noise]
CALLER: smart electric
CALLER: [noise]
AGENT: [noise]
AGENT: [cough]
AGENT: and what is the company address
CALLER: zero two two main street
AGENT: [noise]
AGENT: [noise]
AGENT: [noise]
CALLER: forest ranch oregon
AGENT: [noise]
AGENT: [noise]
CALLER: eight six seven
AGENT: [noise]
AGENT: [noise]
CALLER: five one
AGENT: [noise]
AGENT: [noise] [noise] [noise]
CALLER: [noise]
AGENT: can you please repeat the zip code
AGENT: [noise]
CALLER: i'm sorry what
AGENT: can you please repeat the zip code [noise] okay and what is the bill amount
CALLER: [noise]
CALLER: eight seven two six one
AGENT: [noise]
CALLER: eighty seven dollars
CALLER: [noise]
AGENT: k we will send your payment to smart electric is there anything else i can help you with
CALLER: [noise]
CALLER: no that's all thank you
AGENT: thank you for calling have a great day
CALLER: [noise]

Figure 10: Supplemental Example 6

AGENT: hello this is harper valley national bank my name is mary how can i help you today
CALLER: hi my name is elizabeth smith
CALLER: i need to check my account balance
AGENT: sure i can help you with that
AGENT: uh what which account would you like to check
CALLER: [noise]
CALLER: my checking account
AGENT: sure um so it shows that your checking account balance is sixty seven dollars is there anything else i can help you with
CALLER: no thank you
AGENT: thank you for calling have a great day
CALLER: [noise]
CALLER: [noise]

Figure 11: Supplemental Example 7

AGENT: hello this is harper valley national bank my name is robert how can i help you today
CALLER: hi my name is patricia jones
CALLER: i would like to reset my password
AGENT: okay let's get that set for you and what is your phone number
CALLER: [noise]
CALLER: my phone number is six four nine
CALLER: six seven one
CALLER: one nine nine eight
AGENT: okay
AGENT: your password link has been sent to your email is there anything else i can help you with
CALLER: no thank you
AGENT: [unk] thank you for calling you have a great day
CALLER: [noise]

Figure 12: Supplemental Example 8

AGENT: hello this is harper valley national bank my name is elizabeth how can i help you today
CALLER: [noise]
CALLER: hi my name is david wilson i would like to pay a bill
AGENT: [noise]
AGENT: [noise]
AGENT: [noise]
AGENT: what's the company name
CALLER: fossil gas
AGENT: [noise]
AGENT: [noise]
AGENT: [music]
AGENT: okay and what's the company address
CALLER: the address is eight zero three
AGENT: [noise]
CALLER: main street
AGENT: [noise]
AGENT: [noise]
CALLER: that's in harper valley oregon
AGENT: [noise]
CALLER: nine five seven five eight
AGENT: [noise]
AGENT: could you please repeat the zip code
CALLER: nine five seven five eight
AGENT: [noise]
AGENT: [noise]
AGENT: and what's the bill amount
CALLER: the amount of the bill is ninety two dollars
AGENT: [noise]
AGENT: [noise]
CALLER: [noise]
AGENT: okay
AGENT: i have sent your um bill to fossil gas is there anything else i can help you with today
CALLER: that will be all thank you
AGENT: [noise]
AGENT: thank you for calling have a good day
CALLER: [noise]
AGENT: [noise]

Figure 13: Supplemental Example 9

AGENT: hello this is harper valley national bank my name is mary how can i help you today
CALLER: [noise]
CALLER: [noise]
CALLER: um hi my name is mary davis uh i need i will like to have a new check sent to my home address
AGENT: okay a new checkbook
CALLER: [noise] yeah
AGENT: what is your address
CALLER: so my address is three four five main street
CALLER: the city is harper valley
CALLER: and the state is california
CALLER: and the zip is nine one two four nine
CALLER: [noise]
CALLER: [noise]
AGENT: okay
AGENT: can you repeat that zip code
CALLER: yes it's nine one two four nine
AGENT: nine one four two nine
CALLER: no nine one
CALLER: two four nine
AGENT: okay nine one
AGENT: two four nine
CALLER: yeah
AGENT: okay
AGENT: i have
CALLER: [noise]
AGENT: three four five main street
CALLER: yes
AGENT: harper valley
CALLER: and
AGENT: california
CALLER: yes harper valley yes
CALLER: [noise]
AGENT: nine one two four nine
CALLER: four nine yes
CALLER: [noise]
AGENT: okay a new check book has been sent to your home address is there anything else i can help you with
CALLER: uh no that's all i need for today thank you
AGENT: alright thank you for calling have a great day
CALLER: you too bye
AGENT: mmm bye

Figure 14: Supplemental Example 10