Auditory Scene Recognition Using Textual Knowledge

ABSTRACT

Auditory scene recognition is the process of recognizing the context of audio clips from a set of predefined class labels. In this paper, we introduce a new event based scene recognition framework. Different from traditional approaches which train statistical models from samples of the same set of scenes, our framework doesn't require any such labeled training data. Instead, using a comprehensive concept knowledge base, we are able to develop a vocabulary of audible event terms and use them to query for shorter audio samples of such primitive events from the web. The framework then automatically mines the event-scene distribution from large amount of online TV or movie transcripts. Combined with the Gaussian mixture model learned from the event samples, this event-scene probability model allows us to automatically infer the most likely scene of any input audio. Despite that no training samples of the original scenes are used, the framework achieves 10-scene classification accuracy closely on par with the best-performing machine learning technique reported by the latest IEEE AASP scene recognition challenge in 2013.

Keywords

Gaussian Mixture Model, Audio Scene Recognition, Knowledge Base, Clustering

1. INTRODUCTION

Auditory scene recognition (ASR) [39] is the problem of classifying an audio sample into one of a set of predefined complex scenes, each of which may comprise of multiple simple sound events. Examples of such scenes include "street", "supermarket", "restaurant", "park", etc. A "street" scene might include simple sound events such as engine noises, car honking, sirens, people walking, and traffic signal beeps.

ASR finds many applications in mobile devices, robotics, criminal investigation and national security. For example, with ASR, a cell phone can detect that it is in a meeting and automatically turn down the volume; or plays appropriate music when in different environment to match the activity or mood of the user. Another example is crime investigation. Law enforcement team can use ASR to recognize the background context of audio recordings from wire-tapping, phone conversations, and other sources. In the past, it takes human experts many hours to repeatedly sieve through many audio recordings with ad hoc accuracy. ASR automates this process and significantly improves the effeciency of crime solving.

ASR is an interesting problem because i) audio sensors are one of the most inexpensive sensors to build and deploy and they are already deployed in almost every smart phone; ii) these sensors as well as conventional audio recording have created a large amount of digitized audio information and much of this is readily available online; iii) the amount of information in even high-quality audio information is much smaller than images and videos and hence it is easier to store and process; and iv) the past century has seen remarkable progress in digital signal processing and there are many mathematical tools for (pre)processing audio signals (either mono or stereo). For these reasons, it appears that we have the necessary ingredients for solving this important problem.

However, ASR still presents a few major challenges. First, even to human beings, recognizing a scene from an audio clip is not an easy task. Peltonen *et al.* [40] remarked that the best humans can do for classifying into 25 different scenes is just 70%. Second, even though a number of successful techniques have been developed for speech recognition, these techniques cannot be directly applied to auditory scene recognition, despite their similarities, because human speech is made up of limited number of phonemes as basic units, whereas environmental scenes have much larger variations. Thus, understanding and recognizing environmental audio scenes is a much harder problem. Third, while it is well known that humans recognize scenes by detecting simple events [40, 24], it is still difficult to separate the events from a mixture of signals or sources [13]. Therefore, most machine learning based approaches do not attempt to recognize the simple events in an input sample, but instead train models of the scenes directly from samples. Such approaches have so far met limited success (with best accuracy close to humans [19]) because a complex scene can have so many variations that large number of labeled training samples are required to build an accurate model; but such training samples can be hard to obtain.

In this paper, we adopt a big-data, knowledge driven approach in which we derive knowledge about the relationships between a scene and its constituent events from large text corpus and comprehensive concept and word taxonomies. This allows us to building *scene-event* mappings for virtually any environmental scenes without supervised training. Then separately, we can train auditory detection

models for each primitive events such as car honks and dog barks, from audio clips from the web. With these primitive models, we will then be able to detect the probability distributions of events in an input audio sample, and thus infer the likelihood of a particular scene according to the scene-event map. For this paper, we focus our attention on mono audio samples, but techniques developed here can be readily adapted to stereo sounds.

The main contributions of our work are:

- This paper is first-of-its-kind research which combines text mining with audio event detection into an unsupervised auditory scene recognition framework that requires no human intervention. Therefore it can be used to handle large number of scenes and scenes for which very few audio samples are available (Section 3).
- We leverage a large number of online movie and TV scripts to train probabilistic models of common scenes based on audible concepts. This approach can scale up to very complex and unusual scenes. While we showcase this technique using movie and drama scripts, more scenes can be modeled from other textual corpuses such as novels, news and even the general web pages (Section 3.1 and Section 3.2).
- We propose a method to cluster audio event training data where each cluster represents a particular type of that even, e.g., a certain species of dog for the "dog" event, or a certain aspect of an event. The clustering approach effectively removes noises in the training data and improves the quality of the models for the primitive events (Section 3.3).
- Our ASR framework achieves average accuracy of 42% on a 10-scene classification data set. This is on par with the best performing method from AASP Challenge using training data of the scenes themselves (Section 5).

2. THE ASR PROBLEM

The input of the ASR problem is a digital audio clip A, and a set of terms S that describe different auditory scenes, such as "street", "supermarket", "train station", etc. The output of the problem is a classification label $l \in S$ which best characterizes the context or scene under which A was recorded. This is a typical multi-class classification problem.

An audio clip in this paper is a single channel, monaural discrete signal wave such as in Figure 1. Stereo sound can be merged into mono sound and become input to our problem as well.

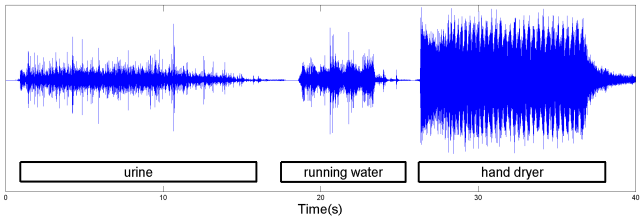

Figure 1: A waveform of an audio clip recorded in a toilet

Traditional machine learning approach to solving ASR is to first obtain audio clip samples labeled with scenes from S, and then learn statistical models from audio features extracted from the training samples. However, because auditory scenes can be complex and come in large variety, large number of labeled samples are required to train an accurate model. What's more the clips are required to be long enough to accommodate sufficient features, hence both the training and storage costs are high. For uncommon scenes, acceptable training samples are hard to obtain.

Table 1: Possible Audible Event Terms from Probase

Concepts	Entities
sound	barking dog, music, *classical*
noise	siren, traffic, *light*
animal	dog, cat, snake
sound effect	chorus, gunshot, *delay*
musical instrument	guitar, oboe, trumpet

Previous research has shown that, human beings recognize auditory scenes not only by global features but, more critically, by detecting important events associated with the scene. For the "toilet" scene in Figure 1, the distinctive audible events or objects that are often detected by humans are "urine", "running water" and "hand dryer." The advantage of recognizing auditory scenes by their constituent events, is that these basic events usually has shorter durations thus more training samples available, and are relatively easier to train. The goal of this paper is to follow this exact intuition, and infer auditory scenes without training samples of the scenes. The key challenges would be i) detecting the sound events from long audio inputs, and ii) relating basic sound events to the correct scene. We do this by a hybrid technique that combines text mining with audio signal modeling.

3. APPROACH

Our ASR framework can be roughly divided into two parts: the *text modeling* and the *audio modeling*. In the text modeling part, we seek to derive probability distribution of predefined auditory scenes given a primitive audible event concepts. For example, the distribution of event "car" may be

$$\begin{aligned} Pr(street|car) &= 0.6 \\ Pr(station|car) &= 0.2 \\ Pr(park|car) &= 0.18 \\ Pr(cafe|car) &= 0.02 \end{aligned}$$

We obtain such probability distribution by first collecting a vacabulary of audible event concepts such as "car honk" and "engine", and then by mining the relationships between these concepts and the scene terms from large volume of text corpus, in particular, movie and TV drama transcripts. In the audio modeling part, we first download audio samples of all audible events from our vacabulary and then train Gaussian Mixture Models (GMMs) for each event using the corresponding samples. During the end-to-end scene recognition phase, the input audio clip is segmented into pieces, and passed to an inference engine, which infers the probability distribution on events for each segment. Finally, based on the event-scene relations obtained in the text modeling part, the engine determines the most likely scenes. Figure 2 shows an overview of our system. Next, we describe the different components of our framework.

3.1 Build a Vocabulary of Audible Concepts

We create the audible concept vocabulary by a bootstrapping iterative process. Each iteration involves a "growing phase" which en-

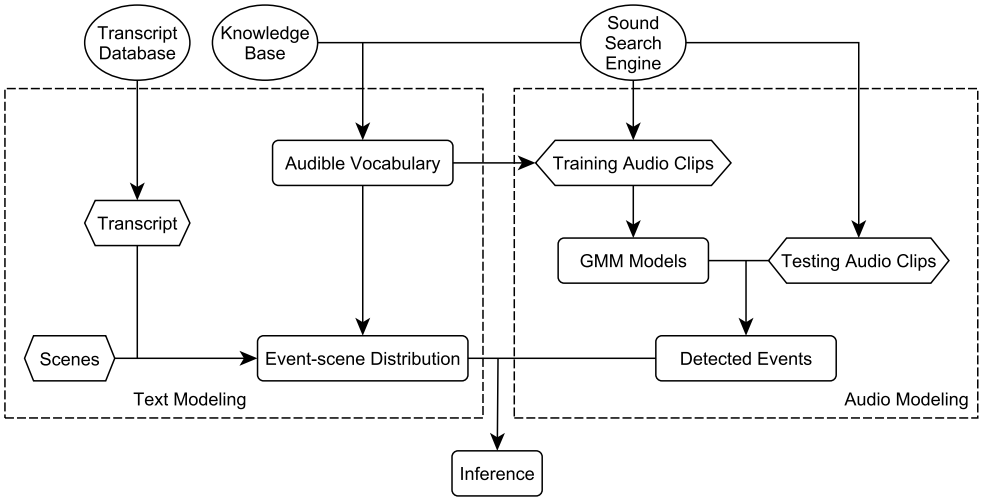

Figure 2: The Auditory Scene Recognition Framework

large the current pool of audible concepts by including additional terms from both an online sound search engine and a knowledge base, and a "filtering phase" which removes some of the terms which are deemed inaudible from the current pool, using the same sound search engine. The iterations stop when no new terms can be added after the filter phase. The final pool of concepts become the vocabulary of audible events.

The knowledge base we use for this purpose is called Probase[44], which is a probabilitic taxonomy of terms organized in hypernymy-hyponymy (isA) relations[1]. Each isA pair (c, e)[2] is associated with a frequency which is the number of evidences that support this isA relation in a large text corpus, and two probability scores known as typicality, defined by $P(e|c)$ and $P(c|e)$, which are calculated from the statistics of the occurrences of terms e and c in the corpus.

We start the bootstrapping process by creating an initial pool of seed candidate event terms. These initial terms were the k most typical hyponyms (by typicality $P(e|c)$) under the terms such as "sound", "noise", "musical instruments", etc. Table 1 gives the some examples of these candidate audible events terms discovered from Probase. One can see that not all of these terms are truly audible events (those italicized terms in the table). We will remove such noises in the later filtering phase.

In the **growing phase**, we enrich the current pool by adding related terms from two sources. We first query a sound search engine for each existing terms in the pool. The resulting clips for each query (e.g., "hunt dog") carry tags such as "labrador" and "puppy". All such terms which exist in Probase as entities (i.e., as e in an isA pair) and are not already in the current pool are considered new candidates. We further expand the set of new candidates by clustering them under different super-concepts. During clustering, we represent each new term as a vector if its super-concepts in Probase and compute distance between any two by Cosine similarity. This way, we could group different variants of dogs together under the concept "dog". Since these variants are probably audible, we deduce that other entities under "dog" are also audible, and therefore add the most typical entities under "dog" which are not in the pool as new candidate terms as well. Figure 3 illustrate this process.

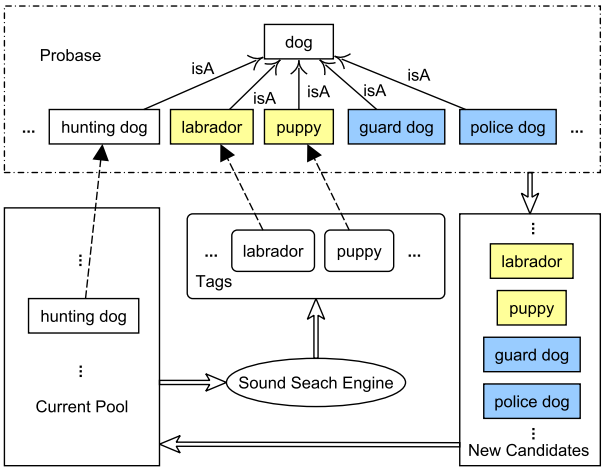

Figure 3: Expansion using Probase

In the **filtering phase**, every new candidate term from this iteration is searched in the sound search engine. We look at the information of the returned audio clips for each term. All clips which are shorter than 0.1 seconds or longer than 30 seconds are removed, because these are usually not a single event by our experience. Finally we filter out terms which have fewer than 10 resulting clips, and keep the rest in our pool and go on to the next iteration.

3.2 Build Event-Scene Probability Model

This section decribes how to compute the conditional probability $P(scene|event)$, for the given n scenes in question and all terms in the event vocabulary. We first extract fragments of text corresponding to the input scenes from large number of TV drama or

[1]Hypernymy relation, also known as concept-entity relation, is the most important relation in Probase, but there are other relations as well.

[2]Here c stands for a concept and e stands for an entity and the two are related by isA relation: e isA c.

movie transcripts, and then extract audible events from the text. Finally we compute the probability model between events and scenes.

3.2.1 Extraction of Scene Contexts

Because movies and TV dramas contain large number of scenes which coincide with the audio scenes of interest, in this paper, we choose to use these transcripts as our primary source to obtain the event-scene relations. Table 2 shows a subset of dramas and movies which were considered as data sources[3].

Table 2: Selected Movies/TV Transcripts

Title	Type	Length
The Big Bang Theory	TV	126 episodes
Friends	TV	229 episodes
How I Met Your Mother	TV	135 episodes
Prison Break	TV	23 episodes
Lost	TV	118 episodes
Sherlock	TV	6 episodes
Family Guy	TV	104 episodes
South Park	TV	232 episodes
Arrested Development	TV	22 episodes
Scrubs	TV	150 episodes
Modern Family	TV	84 episodes
House M.D.	TV	177 episodes
Supernatural	TV	167 episodes
The Vampire Diaries	TV	82 episodes
Firefly	TV	11 episodes
True Blood	TV	34 episodes
Seinfeld	TV	179 episodes
Wall-E	Movie	97 minutes
V for Vendetta	Movie	132 minutes
Twilight	Movie	121 minutes
Toy Story	Movie	81 minutes
Titanic	Movie	194 minutes
Kung Fu Panda	Movie	92 minutes
King-Kong	Movie	187 minutes
I am Sam	Movie	132 minutes
The Avengers	Movie	142 minutes
Avatar	Movie	162 minutes
2012	Movie	158 minutes
500 Days Of Summer	Movie	95 minutes
E.T.	Movie	115 minutes

... Elliott and Mike walk down the driveway.
They are on their way to school.
They discuss E.T., arguing about how smart he is.

[This is just a transition scene.]

EXT: STREET: DAY

Mike and Elliot walk towards a bus stop where a group of children are waiting.
Mike's friends torment Elliott about his "goblin." ...

Figure 4: A Snippet from the Transcript of Movie E.T.

The advantage of using drama transcripts here is that most of them have clear indications of entering or leaving a scene such as in Figure 4. We can make use of such patterns to extract the text context of a scene such as "street." as well as its noun synonyms, e.g., "avenue" and "boulevard.", from WordNet [33]. In addition, because event terms that occur in human conversations are not necessarily events that happen in that scene, we remove all conversations which also have clear patterns from the context.

[3]Movie scripts were downloaded from `http://www.imsdb.com`, while TV series transcripts were downloaded from `http://simplyscripts.com/tv.html`.

3.2.2 Extraction of Events from Contexts

Contexts extracted in Figure 4 may contain event terms such as "walk", "bus" and "children" that find exact match in our vocabulary. The vocabulary also contains compound terms such as "open door", "ring bell" which may not find exact match in the context. To extract as many events as possible from the context, besides exact matches for terms in the vocabulary, we also parse the context using a dependency parser, and pay special attention to the following relations: *direct object*, *indirect object*, *noun compound modifier*, *nominal subject* and *passive nominal subject*. Each of these relations relates either a noun and a verb, or between two nouns. The reason we focus on these dependencies is that sound is generally made by a motion or action and its agent or recipient (single verb or verb-noun cases) or some object (single noun or noun-noun cases such as "coffee cup") alone. A word pair (w_1, w_2) with the above five relations from the context is considered an audible event, if there is a compound event term w_1w_2 or w_2w_1 in the vocabulary.

3.2.3 Event-scene Distribution

Our problem is to classify an input audio clip into one of n predefined scenes. The intuition is that humans recognize a scene by its most important, and distinctive events. We model this by $P(scene|event)$. For example, if flushing the toilet is a very distinctive event for the scene "toilet", then we expect $Pr(toilet|flush)$ is significantly higher than $Pr(other\_scene|flush)$.

We compute the probability as

$$Pr(scene = s|event = e) = \frac{TF(s,e)}{\sum_{s \in S} TF(s,e)}, \qquad (1)$$

where s is a scene in the set of n input scenes S, e is an event in the vocabulary, and $TF(s,e)$ is the number of occurrences of e in the context of scene s.

3.3 Train Models for Audible Events

Once we download audio samples (details in Section 4.3) for each of the event terms in the vocabulary, we can train audio models for each event.

3.3.1 Feature selection

To train audio models, first we need to find a good way (features) to describe the audio data. We view an audio clip as a sequence of logically overlapping frames, each comprised of fix number of sample points, as in Figure 5. The amount of overlap is a parameter of the model. A frame is the basic unit of feature extraction, in either time or frequency domain.

Existing works[19, 4, 34, 46] reported that, in the frequency domain, the mel-frequency cepstrum coefficients (MFCC) feature is a widely-used feature which is a cepstral representation of the audio clip, i.e., a non-linear spectrum-of-a-spectrum. It is fairly robust because it closely resembles the human auditory system's response to different frequency bands.

In the time domain, short-time energy is the energy measure of a short segment of sound. It has been shown to be a simple and effective feature to distinguish active voice against silence in speech processing[18].

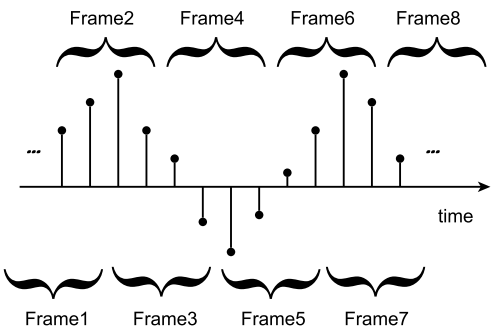

Figure 5: Overlapping Audio Frames

Zero crossing rate (ZCR) is a temporal feature which measures the rate at which the signal transits from positive to negative or back. ZCR can be used to distinguish meaningful events from environmental noise, since environmental noise usually has a larger ZCR.

In our work, we use short time energy to remove ambient noises first. Then, we combine ZCR and MFCC as features, which will be used in the audible events modeling next.

3.3.2 Event models

In this paper, every audible event in the vocabulary is modeled as one or more GMMs. A GMM is a weighted sum of M Gaussian density function, which is given by:

$$P(\mathbf{x}|\mathbf{\Theta}) = \sum_{i=0}^{M-1} c_i \prod_{d=0}^{D-1} \frac{1}{\sqrt{(2\pi)}\sigma_{i,d}} e^{-\frac{1}{2\sigma_{i,d}^2}(x_d-\mu_{i,d})^2}, \quad (2)$$

where $\mathbf{x}$ is a D-dimensional variable representing the feature vector, Θ is the parameters of GMM, including $\mathbf{c}$, μ and σ. c_i is the weight of the i^{th} mixture, with the following constraint:

$$\sum_{i=0}^{M-1} c_i = 1, \quad (3)$$

while $\mu_{i,d}$ and $\sigma_{i,d}$ are the mean and standard deviation of the dimension d of the i^{th} mixture.

3.3.3 Model training

The audio samples downloaded for each event term in the vocabulary, such as "dog" may sound very different, either because there are various aspects of an event, or in the case of "dog", simply because there are different species – *bull dogs* certainly sound very different from *chihuahuas*. To accurately models each event, in this paper, we aim to derive multiple GMMs, each for a separate aspect of an event.

Before we do the actual training, we first remove ambient noise from the training clips. We compute the short time energy for each frame of the sample:

$$\overline{E} = \frac{1}{N}\sum_{i=0}^{N-1} x^2(i), \quad (4)$$

where N is the number of sample points in a frame, and $x(i)$ is the value of i^{th} sample point. We only retain the frames with energy higher than average among all frames. The contiguous frames after the noise removal become segments within the event. We then remove tiny segments which are shorter than 100ms, and further split longer segments into 500ms-long pieces. We believe these resulting segments may carry different aspects of the same event.

Next we put all remaining segments from all audio samples of the same event together and cluster them. We train an interim GMM for each segment and use KL divergence [29] as a distance measure for clustering.

The GMM for each segment is trained by EM algorithm:

$$P(\mathbf{O}|\mathbf{\Theta}) = \prod_{i=0}^{L-1} P(\mathbf{O_i}|\mathbf{\Theta}). \quad (5)$$

where $\mathbf{O_i}$ is the feature vector for frame i combining ZRC and MFCC features, and L is the number of frames in the segment. ZRC is calculated as:

$$\overline{Z} = \frac{1}{2(N-1)}\sum_{i=0}^{N-2}(|sgn(x(i)) - sgn(x(i+1))|), \quad (6)$$

where

$$sgn(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}, \quad (7)$$

and N and $x(i)$ carry the same meaning as Equation (4).

KL divergence measures the difference between two probability distributions:

$$KL(P||Q) = \int_{-\infty}^{+\infty} \ln(\frac{P(x)}{Q(x)})P(x)\mathrm{d}x, \quad (8)$$

We estimate the integration in Equation (8) as follows to calculate KL divergence between two segments A and B:

$$KL(A||B) = \frac{1}{n}\sum_{i=0}^{L-1}(\ln P(\mathbf{a_i}|\mathbf{\Theta_A}) - \ln P(\mathbf{b_i}|\mathbf{\Theta_B})), \quad (9)$$

where L is the number of frames of in segment A and B, $\mathbf{a_i}$ is the i^{th} frame of A, $\mathbf{b_i}$ is the i^{th} frame of B, and $\mathbf{\Theta_A}$ and $\mathbf{\Theta_B}$ are the GMM parameters of segment A and B, respectively.

After clusters of segments are formed, we group the segments within a cluster together and re-train a GMM using the features from the combined segment for each cluster. As such, each event is associated with one or more GMMs, each for a unique aspect of this event.

3.4 Scene Inference

To classify a new audio clips into one of the scenes, we segment the input clip in the same way as we did for training samples. For each segment, we compute the posterior probability of a segment seg_i given an even e as

$$P(seg_i|e) = \sum_{e_j \in e} P(seg_i|\Theta_{e_j}). \quad (10)$$

where e_j is an aspect of e. In order to reduce the complexity, we only consider top K aspects for the event, for each segment. Then the final score for a scene s is:

$$score(s) = \sum_{i,e}(P(seg_i|e) \times len(seg_i) \times P(s|e)), \quad (11)$$

where $len(seg_i)$ is the length (number of frames) of segment i, and $P(s|e)$ is given by Equation (1). The scene with the highest score is the most likely scene for the audio clip.

4. IMPLEMENTATION AND DISCUSSION

This section discusses a few implementation details which are necessary for creating a prototype system and makes some additional remarks about our approach.

4.1 Expansion of Audible Event Terms

In the growing phase of the vocabulary construction process, we expand the candidate pool by collecting tags or filenames of the returned results from the sound search engine. Our source of event terms as well as audio clips come from the following sources:

- FreeSound: `http://www.freesound.org`
- SoundJax: `http://soundjax.com`
- FindSounds: `http://www.findsounds.com`
- MediaCollege: `http://www.mediacollege.com`
- SoundRangers: `http://www.soundrangers.com`

The tags or keywords obtained from these engines can be noisy. We thus lemmatize the words and remove redundant words like "... noise", "... sound effects", "sound of ...", "... ambience". For example, we transform "the sound of barking dog" into "barking dog", "churr sound" into "churr", before matching them in Probase. The reason we require all these new terms to be Probase entities because we hope to find their conceptual siblings in the taxonomy to further expand the vocabulary.

We are conservative about adding siblings of an existing candidate event into the pool. For example, if we already have "dog", "cat" and "donkey" in the pool, we may be able to deduce that "animal" is their common super-concept by clustering. However, many entities under "animal", such as "oyster" make virtually no sound. We only admit entities of a concept c into the pool if the proportion of entities in c that are already in the pool is larger than a threshold, say 0.5.

Building the audible event vocabulary is one of the most critical steps in this work. While Probase and the sound search engine provide indications of whether a term is audible, they are not always reliable. Also, the current approach restricts the vocabulary to terms which are in Probase, which means some audible event terms might be excluded.

4.2 Collection of Text Contexts

When extracting contexts for scenes in transcripts, we used a stricter approach that guarantees the text segment being extracted indeed describes the scene, because we match the scene word or its synonyms with open scene patterns in the scripts. This, however, can lead to insufficient number of contexts and thus unreliable event-scene distributions or biases. Another approach is to extract scene texts which *contains* the scene word or its synonyms. We didn't implement our system this way because while it improves the coverage, it brings about much more noise. The volume of our corpus is not statistically large enough to reflect the true dependency between the scene words and associated audible event terms. This statistical significance, however, may be achieved by mining the relations between the two on much larger web corpus using for example simple co-occurrences. This is a possible direction of future work.

4.3 Training Models for Events

We query every terms in the vocabulary in a sound search engine.[4] Because the engine does fuzzy matching, not all clips returned are about the event searched. Therefore we only keep those clips whose title contain the original query term or all of its contituent words, after lemmatization. Note that We do not perform synonym matching here because the terms in the vocabulary maybe synonyms but each of them will be associated with a set of audio samples.

In the current set-up, models are trained for every audible event term, even though they are similar to each other, like "guard dog" and "police dog". The training process can be lengthy despite that no annotation is required. It appears that a balanced taxonomy of sound effects could partially ameliorate the problem, and create more balanced trained models. Similar attempts have been made to create such taxonomies around WordNet for both sounds [7] and images [16]. Nevertheless, one must be reminded that the audio model training only needs to be done once for each audible event, and the models can be reused for classifying into different set of scenes.

4.4 HMM vs. GMM

Besides GMM, left-to-right Hidden Markov Model has also been widely used in speech recognition, where each hidden state of the HMM is a GMM. HMM is not a good choice for event detection for these reasons:

- The quality of training samples for environmental sound is generally not as good as training samples for speech recognition.
- For a given event, for example, *dog barking*, the training sample may repeated several dog barks. It is not easy to break it up into a sequence of single dog barks. Although we can model the event as a cyclic HMM, it does not perform well in practice.
- For some of audio events, interestingly, if we reverse the audio sample and play backwards, it can still be recognized by human beings. Thus, markov process is not effective in describing such audio events.
- HMM is a complex model, and can lead to overfitting when we do not have enough training data for some events.

5. EVALUATION

We evaluate our ASR system by comparing it with the official baseline system using GMM (called Base) and a best performer using SVM (called Best) from the IEEE AASP scene classification challenge [11] on the accuracy of classifying up to 10 scenes. In addition, we show the most popular sound events detected by our system for each of these scenes.

[4]We primarily use FreeSound.org as it provides better tags and the quality of its clips are generally better.

5.1 Experiment Setup

Our dataset consists of audio samples of 10 scenes: **bar**, **beach**, **cafeteria**, **church**, **concert**, **office**, **park**, **street**, **toilet** and **train**. There are 10 training clips (for Base and Best) and 10 testing clips for each scene.[5] The duration of every training clip ranges from a few seconds to a few minutes, while all the testing clips are truncated to 20 seconds long. All clips are converted to WAVE form with a single channel, 16 bits per sample point, 384kbps bit-rate. We did not use the IEEE AASP data because those clips generally do not contain detectable events even to the human ear. Instead they carry global features which are good for traditional ASR modeling. Moreover, the training and test data appear to be similar to each other, giving extra advantage to Base and Best.

We set the frame size to 512 sample points (20ms) with 50% overlap to extract audio features. The MFCC feature are extracted using open source code written by Klautau in 2001. The implementation is based on [14, 41, 26, 8]. We combine ZCR, 12-dimensional MFCC and its 1st- and 2nd-order differentials (37 dimensions in total) to train our GMM models. The number of mixtures in GMM is set to 32.

All experiments are conducted on an 4-core[6] Intel Core i7 3.4GHz desktop computer with 32GB memory, running Windows Server 2012.

5.2 Event Detection

Our audible event vocabulary contains 2326 terms. Total number of contexts extracted from the script corpus for the 10 scenes is 4953. Because not all these events are significant, to simplify training, we only keep those events which occur sufficient number of times (0.01% out of all event terms) in at least one of the 10 scenes in our text corpus. As a result, 184 acceptable events were found the in these 10 scenes in the corpus and these are used to build the even-scene distribution.

Table 3 gives 5 most probable events detected for each of the scenes. Most of the scenes successfully have their important events detected in the test samples. However, some false positive events present, such as "bark" which exists in many scenes. The reason is "bark" is a very popular keyword on sound search engines which return many diverse, noisy audio samples. Consequently, the model for "bark" is relatively coarse and thus can be erroneously detected almost everywhere. "Station" is also detected incorrectly at several places. This is because "station" actually is more of a scene than a single event, since it is represented by a global ambience with noises, vague human speech and perhaps music in the backgound. Such global features are present in public areas. "Urine" was not detected in the "toilet" scene because our samples don't have it.

5.3 Classification Accuracy

Table 4 compares the recognition accuracy for each audio scene between Base, Best and two of our variants. Classification accuracy is defined as

$$\text{Accuracy} = \frac{\text{Number of correct labels}}{\text{Total number of labels}}$$

[5]These clips are obtained by querying the scene words on FreeSound.org and are available at `http://202.120.38.146/~kzhu/audio/`.

[6]All our programs are single-threaded, so only one core is utilized at a time.

Table 3: Top 5 Detected Events per Scene

Scene	Events				
street	traffic	engine	station	bark	subway
office	bark	dish	bike	backpack	bathroom
beach	sea	water	sand	river	shore
concert	applause	bark	tv	tunnel	cello
cafeteria	sea	subway	backpack	dish	engine
bar	pub	kid	station	bark	angel
church	bell	chant	match	cracker	ghost
train	station	tower	bathtub	carriage	kid
toilet	water	pill	devil	dish	bath
park	traffic	bark	engine	boat	bus

Here "top 1" and "top 3" refer to the correct label found in most probable scene or in the top 3 most probable scenes. We obtain an average accuracy of 42% for "top 1" and 67% for "top 3" in our experiment.

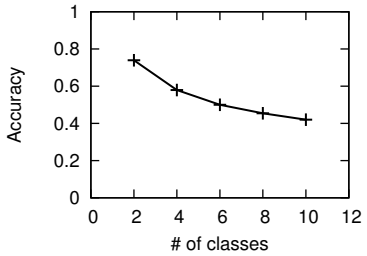

Figure 6: Recognition Accuracy vs. Number of Classes to Recognize

We then investigate how the number of scene classes affects the recognition accuracy. We re-run the training and testing phases for our system for every combination of 2 classes, 4 classes, 6 classes and 8 classes and calculate the average accuracy for each 4 cases and plot a graph in Figure 6. As expected, the accuracy is almost 80% for 2-class recognition, and gradually decreases as the number of classes goes up.

A common metric in visualizing the quality of N-way classification method is the confusion matrix, which shows the number of times a test sample is classified into each of the N classes. Ideally, high numbers should be on the diagonal of the matrix. From this matrix, We can see that some of scenes for which the events are correctly detected (such as "beach", "street" and "train") are recognized with very high accuracy, while other scenes such as office do not have such good luck.

6. RELATED WORK

ASR and related problem has been extensively studied in the past. Some researchers work on sound event detection/classification/recognition, which focuses on some specific sound events, ignoring the contexts or scenes that produce the events. Other researchers work on recognizing the contexts or scenes, but most of them use one model for recognizing the context without detecting the sound events therein. Recently, there has been some new efforts to use a small set of pre-defined event to infer the context [24], which partly inspires this paper's research.

6.1 Auditory Scene Recognition

Table 4: Recognition Accuracy for 10 Audio Scenes

	bar	beach	cafeteria	church	concert	office	park	street	toilet	train
Ours (top 1)	10%	**100%**	30%	40%	50%	0%	10%	30%	70%	**80%**
Ours (top 3)	30%	**100%**	70%	**80%**	**80%**	10%	50%	**80%**	**90%**	**80%**
Baseline	20%	40%	50%	40%	30%	10%	30%	60%	50%	30%
Best	60%	10%	40%	70%	60%	70%	10%	50%	40%	30%

Table 5: Confusion Matrix of 10 Scene Recognition

	bar	beach	cafeteria	church	concert	office	park	street	toilet	train
bar	1	1	2	0	2	0	0	2	0	2
beach	0	10	0	0	0	0	0	0	0	0
cafeteria	0	3	3	0	1	0	0	0	0	3
church	1	0	1	4	2	0	0	0	0	2
concert	1	0	1	0	5	0	0	0	0	3
office	0	1	4	0	2	0	1	0	1	1
park	1	2	2	1	1	0	1	0	0	2
street	0	0	2	0	1	0	0	3	0	4
toilet	0	1	1	1	0	0	0	0	7	0
train	0	0	0	1	0	0	0	0	1	8

Gaunard *et al.* [21] proposed an HMM-based environmental noise recognition system. It use discrete HMM as model and linear prediction cepstral coefficients (LPCCs) as features. It shows that the system can achieve 95.3% correct rate, which outperforms human listeners who only achieves 91.8% correct rate for classifying 5 types of environmental noise (car, truck, moped, aircraft, and train).

Peltonen *et al.* [39] studied the efficiency of different acoustic features, models, and the effect of test sequence length. They proposed two system using different features and models. One was using band-energy ratio as features and trained by 1-NN classifier. The other was using MFCC as features and trained by GMM. The best recognition rate is around 68.4% for 26 different scenes.

Eronen *et al.* [19] investigated the feasibility of an audio-based context recognition system. It showed that linear data-driven transformations, *i.e.* Independent Component Analysis (ICA) and Linear Discriminant Analysis (LDA) could improve recognition accuracy slightly. Their system can achieve 58% accuracy for 24 common contexts. They also did some listening tests, and found that human beings can achieve 69% accuracy on the same data set.

Chu *et al.* [10] performed an empirical feature analysis and use the matching pursuit (MP) algorithm to obtain effective features from a large feature set, including MFCC, LPCC, band energy ratio, zero-crossing, energy, etc. The recognition rate of their system is 82.3% over 14 audio contexts.

Weninger *et al.* [43] focus on animal vocalizations. They compared left-to-right HMM, cyclic HMM, recurrent neural networks, and SVM, and achieve up to 64.0% accuracy on a 5-class task, and 81.3% on a 2-class task.

One of the approaches to consider audio events in context inference is introduced by Cai *et al.* [4]. They proposed a flexible framework to recognize 5 audio contexts, including excitement, humor, pursuit, fight and air-attack, using 10 predefined events. The main technologies they used are HMM, Grammar Network, and Bayesian network. Their system can achieve 91.7% accuracy for event detection, and 82.4% accuracy for context inference.

Recently, another event-based audio context recognition is proposed by Heittola *et al.* [24]. They use a histogram of audio events which are detected by GMM/HMM presented in [32], where an accuracy of 24% was obtained when classifying isolated sound events into 61 classes. After a histogram of audio events was built, context recognition can be performed by using cosine distance to calculate the similarity. Their system can obtain 89% accuracy when recognizing 10 audio contexts. However, they used predefined set of events, which is equivalent to collecting all the relevant events and their training samples in our paper, only manually!

Giannoulis *et al.* [23] described a public evaluation challenge[7] on acoustic scene classification and detection of sound events within a scene. They provided an overview of systems submitted to the challenge and summarized the results. The challenge is to classify audio clips into 10 different scenes. Chum *et al.* [11] proposed a GMM and HMM based system, using magnitude response, loudness, spectral sparsity and temporal sparsity as features. They achieved accuracy of 72%. Elizalde *et al.* [17] proposed an *i-vector* system[15, 3], together with MFCC features, which can achieve an accuracy of 65.8%. Geiger *et al.* [22] introduced a SVM based system, using many low-level features, such as MFCC, band energy, etc. An accuracy of 73% is achieved by their system using majority voting scheme. Krijnders *et al.* [28] proposed a SVM based system using tonalness as feature, achieved 53% accuracy. Li *et al.* [31] developed a treebagger classifier using MFCC and other spectral features. It can achieve 72% accuracy. Nam *et al.* [35] introduced the feature learning approach to audio scene classification. They use RBM[30]and perform selective max-pooling to form scene-level feature vector for SVM training. Their system can achieve 75% accuracy. Nogueira *et al.* [36] proposed a SVM based system using spectral, temporal and spatial features, achieve accuracy of 69%. Olivetti [37] proposed two approaches, dissimilarity representation and normalized compression distance, to embed audio into a vectorial feature space. A random forest[2] algorithm was used for classification, and the system can achieve accuracy of 80%. Patil *et al.* [38] proposed a framework that provided a analysis of the spectro-temporal modulations in acoustic signal, and built a SVM classifier, which can achieve accuracy of 73%. Rakotomamonjy *et al.* [1] used a constant Q transform in feature extraction. Their system can achieve 75% accuracy by applying a SVM classi-

[7] http://c4dm.eecs.qmul.ac.uk/sceneseventschallenge/

fier. Roma *et al.* [42] proposed a SVM based classifier with MFCC feature and RQA[45] features. It can achieve accuracy of 71%.

6.2 Sound Event Detection

Sound event detection is to detect some pre-defined sound events in a long audio sample. Usually, a large number of labeled audio samples of events are use as training data.

Heittola *et al.* [25] proposed a context-based sound event detection system. They used the ground truth of the context of audio to help them detect the sound events in the audio. They modeled the context using GMM, and the sound events were modeled as 3-state left-to-right HMMs. It is shown that there system can benefit from the context information.

Some work focus on specific sounds, such as gunshots[12], birds[20], etc. Some work focus on context-based sound event detection, which only consider about specific sounds, like kitchen[27], bathroom[9], etc. Their work usually train GMM, HMM or SVM models as classifiers.

We use sound event detection as a part of our work. Instead of directly training models from audio samples, we break the audio into several short segments. Then we cluster those segments and train a GMM for each cluster.

6.3 Audio Processing Using Knowledge

Cano and Koppenberger[5] proposed a solution to automate audio annotation. Sound samples are gathered and are tagged with unambiguous concepts in WordNet. A 20-nearest-neighbor classifier is trained to annotate more audio samples using normalized Manhattan distance. Based on 15 sound effects, an annotation test on 261 audio files showed an accuracy of 91%.

Based on this trial, the authors further built sound effect taxonomy[7] and processed audio retrievals [6] on it. For the taxnomy[7], they implemented a classification scheme for sound effect management on top of WordNet, which solves the ambiguity inherent to natural language. This system both regulates the labels for annotation, and leads to a robust framework for sound information retrieval. Futher, the researchers presented a sound effect retrieval system [6] that incorporates content-based audio techniques and semantic knowledge provided by WordNet.

7. CONCLUSION

In this paper, we present a novel hybrid framework which combines text mining and audio signal processing for recognizing auditory scene. This framework is unsupervised in the sense that no manual labeling of the audio training data is needed. Instead of training audio scene data directly, like most existing work does, we train GMMs on primitive audible events which are downloaded from online sound search engines. Then the framework leverages large text corpus of online TV and movie transcripts to mine statistical models between a scene and its constituent events. Experiments for 10-scene classification showed promising results of 42% accuracy which is higher than the baseline and state-of-the-art methods reported at recent IEEE AASP scene recognition challenge.

8. REFERENCES

[1] G. A. Rakotomamonjy. Histogram of gradients of time-frequency representations for audio scene classification.

[2] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[3] L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matejka, and N. Brummer. Discriminatively trained probabilistic linear discriminant analysis for speaker verification. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 4832–4835, May 2011.

[4] L.-H. Cai, L. Lu, A. Hanjalic, H.-J. Zhang, and L.-H. Cai. A flexible framework for key audio effects detection and auditory context inference. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(3):1026–1039, May 2006.

[5] P. Cano and M. Koppenberger. Automatic sound annotation. In *14th IEEE Workshop on Machine Learning for Signal Processing*, 2004.

[6] P. Cano, M. Koppenberger, S. L. Groux, P. Herrera, J. Ricard, and N. Wack. Knowledge and content-based audio retrieval using wordnet. In *ICETE (3)*, pages 301–308, 2004.

[7] P. Cano, M. Koppenberger, and P. Herrera. Sound effects taxonomy management in production environments. In *AES*, 2004.

[8] R. Cardin, Y. Normandin, and E. Millien. Inter-word coarticulation modeling and mmie training for improved connected digit recognition. In *Proceedings of the 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing: Speech Processing - Volume II*, ICASSP'93, pages 243–246, 1993.

[9] J. Chen, A. Kam, J. Zhang, N. Liu, and L. Shue. Bathroom activity monitoring based on sound. In H.-W. Gellersen, R. Want, and A. Schmidt, editors, *Pervasive Computing*, volume 3468 of *Lecture Notes in Computer Science*, pages 47–61. Springer Berlin Heidelberg, 2005.

[10] S. Chu, S. Narayanan, and C.-C. Kuo. Environmental sound recognition with time-frequency audio features. *Audio, Speech, and Language Processing, IEEE Transactions on*, 17(6):1142–1158, Aug 2009.

[11] M. Chum, A. Habshush, A. Rahman, and C. Sang. IEEE AASP scene classification challenge using hidden markov models and frame based classification.

[12] C. Clavel, T. Ehrette, and G. Richard. Events detection for an audio-based surveillance system. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 1306–1309, July 2005.

[13] P. Comon and C. Jutten. *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press, 2010.

[14] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(4):357–366, Aug 1980.

[15] N. Dehak, R. Dehak, P. Kenny, N. Brümmer, P. Ouellet, and P. Dumouchel. Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification. In *INTERSPEECH*, volume 9, pages 1559–1562, 2009.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[17] B. Elizalde, H. Lei, G. Friedland, and N. Peters. An i-vector

based approach for audio scene detection.

[18] D. Enqing, L. Guizhong, Z. Yatong, and C. Yu. Voice activity detection based on short-time energy and noise spectrum adaptation. In *Signal Processing, 2002 6th International Conference on*, volume 1, pages 464–467 vol.1, Aug 2002.

[19] A. Eronen, V. Peltonen, J. Tuomi, A. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi. Audio-based context recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(1):321–329, Jan 2006.

[20] S. Fagerlund. Bird species recognition using support vector machines. *EURASIP J. Appl. Signal Process.*, 2007(1):64–64, Jan. 2007.

[21] P. Gaunard, C. Mubikangiey, C. Couvreur, and V. Fontaine. Automatic classification of environmental noise events by hidden markov models. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 6, pages 3609–3612 vol.6, May 1998.

[22] J. T. Geiger, B. Schuller, and G. Rigoll. Recognising acoustic scenes with large-scale audio feature extraction and svm.

[23] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. Plumbley. Detection and classification of acoustic scenes and events: An ieee aasp challenge. In *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*, pages 1–4, Oct 2013.

[24] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen. Audio context recognition using audio event histograms. In *18th European Signal Processing Conference*, pages 1272–1276, 2010.

[25] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen. Context-dependent sound event detection. *EURASIP Journal on Audio, Speech and Music Processing*, 2013.

[26] J. Jankowski, C.R., H.-D. Vo, and R. Lippmann. A comparison of signal processing front ends for automatic word recognition. *Speech and Audio Processing, IEEE Transactions on*, 3(4):286–293, Jul 1995.

[27] F. Kraft, R. Malkin, T. Schaaf, and A. Waibel. Temporal ica for classification of acoustic events in a kitchen environment. In *in Proceedings of the INTERSPEECH*, 2005.

[28] J. D. Krijnders and A. Gineke. A tone-fit feature representation for scene classification.

[29] S. Kullback and R. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1), 1951.

[30] H. Lee, C. Ekanadham, and A. Y. Ng. Sparse deep belief net model for visual area v2. In *NIPS*, volume 7, pages 873–880, 2007.

[31] D. Li, J. Tam, and D. Toub. Auditory scene classification using machine learning techniques.

[32] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen. Acoustic event detection in real life recordings. In *18th European Signal Processing Conference*, pages 1267–1271, 2010.

[33] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[34] D. Mitrović, M. Zeppelzauer, and C. Breiteneder. Features for content-based audio retrieval. *Advances in computers*, 78:71–150, 2010.

[35] J. Nam, Z. Hyung, and K. Lee. Acoustic scene classification using sparse feature learning and selective max-pooling by event detection.

[36] W. Nogueira, G. Roma, and P. Herrera. Sound scene identification based on mfcc, binaural features and a support vector machine classifier.

[37] E. Olivetti. The wonders of the normalized compression dissimilarity representation.

[38] K. Patil and M. Elhilali. Multiresolution auditory representations for scene classification.

[39] V. Peltonen, J. Tuomi, A. Klapuri, J. Huopaniemi, and T. Sorsa. Computational auditory scene recognition. In *Acoustics, Speech, and Signal Processing (ICASSP), IEEE International Conference on*, pages 1941–1944, 2001.

[40] V. T. Peltonen, A. J. Eronen, M. P. Parviainen, and A. P. Klapuri. Recognition of everyday auditory scenes: potentials, latencies and cues. *PREPRINTS-AUDIO ENGINEERING SOCIETY*, 2001.

[41] J. Picone. Signal modeling techniques in speech recognition. *Proceedings of the IEEE*, 81(9):1215–1247, Sep 1993.

[42] G. Roma, W. Nogueira, P. Herrera, and R. de Boronat. Recurrence quantification analysis features for auditory scene classification.

[43] F. Weninger and B. Schuller. Audio recognition in the wild: Static and dynamic classification on a real-world database of animal vocalizations. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 337–340, May 2011.

[44] W. Wu, H. Li, H. Wang, and K. Q. Zhu. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 481–492. ACM, 2012.

[45] J. P. Zbilut and C. L. Webber. Recurrence quantification analysis. *Wiley encyclopedia of biomedical engineering*, 2006.

[46] Z. Zeng, X. Li, X. Ma, and Q. Ji. Adaptive context recognition based on audio signal. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4, Dec 2008.