上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

THESIS OF BACHELOR

论文题目：Activity Inference from Audio Signal Using Knowledge Base

学生姓名：李梦璐

学生学号：5090309229

专　　业：计算机科学与技术

指导教师：朱其立

学院(系)：电子信息与电气工程学院

ACTIVITY INFERENCE FROM AUDIO SIGNAL USING KNOWLEDGE BASE

ABSTRACT

Audio activity inference/context recognition means to classification among daily environments using ambient audio clips. In previous works, acoustic events, as basic units, in training clips are manually labeled. This thesis presents a novel method to recognize contexts of audio clips without manual annotation on the training dataset. We first build an audible concept vocabulary, as a definition to audio events that we are concerned, with the help of online sound taxonomies, WordNet and Probase. Short audio clips for these events are then obtained through sound search engines (SSEs), and labeled with their query words automatically. In the training stage, each context is modeled with a set of events that frequently co-occur with it in descriptive corpus. In the testing stage, Mel-frequency cepstrum coefficients (MFCC) of unknown clips are extracted, then individual sound events are detected using a network of Hidden Markov Model (HMM) classifiers with Gaussian mixture models (GMMs). Context recognition is performed by computing the exact similarity between this event set and that of each predefined context. An average classification accuracy of 56% is obtained in the recognition among 10 everyday contexts, while it reaches 72.5% on contexts that have more than 18 important sound events collected. In terms of event detection, the system is capable of recognizing almost half of the events, while the temporal positioning needs further alignment.

Key words: context recognition, Hidden Markov Model, Gaussian mixture models, knowledge base, text co-occurrence

摘要

音频活动推断/情境识别指的是使用特定环境下的录音对录音环境进行分类的过程。在之前的研究工作中，用于训练的音频片段中出现的音频事件需要进行人工标注。本文提出了一种不经过人工标注而进行情景识别的方法。我们首先利用分类学的知识库（WordNet 和 Probase），建立一个可听事件的词汇表，并以此限定了我们所关注的音频事件范围。接下来，我们从音频搜索引擎下载了大量相应的短音频，并以其搜索词为每个音频进行自动标注。在训练模型时，每个情境被表示为在描述性语料库中频繁与之同时出现的事件集合。而在测试阶段，我们为每个音频提取 MFCC 特征，进而通过训练出的（基于高斯混合模型的）隐式马尔可夫模型分类器识别出其中发生过的一系列事件。接下来，我们对比这个待识别事件集合与预先定义的每个情境的事件集合，通过他们共有的事件数判定其相似度。相似度越高，则待识别的录音被认为更有可能是录制于该环境。经过对 100 个音频的测试，系统在识别 10 个不同情境的问题上达到 56%的准确度。其中，对于事件集合规模大于 18 的情境，这个准确度能上升到 72.5%。仅就对于事件的识别而言，系统能够识别出过半的事件，然而事件起止时间的精确度仍然需要改进。

关键词：情景识别，隐式马尔可夫模型，高斯混合模型，知识库，文本共现

Contents

Chapter 1. Introduction

Context recognition/Activity inference is defined as the process of automatically determining the ambient context around a device. Information about the surroundings would enable wearable devices to fit users' instant needs by adjusting the operation mode accordingly. A mobile phone can automatically go into appropriate profile while in a meeting, refusing to receive calls and keeping silent when messages come in. Digital cameras can automatically adjust its shutter speed and aperture size to suit the scene. The roughness in GPS orientation can be compensated with the awareness of context. Portable music players can also switch to playlists customized to that occasion.

Compared with image or video recording, audio captures information from all directions without troubling the user in terms of sensor's (i.e. the microphone's) position and orientation. Besides, audio can provide a richer set of information related to location, activity, or occasion than sensor data collected by accelerometers and gyroscopes.

Using audio signals to infer current activity/context is much similar to speech recognition. However, speech has a natural basic unit, the phoneme, whilst environmental sound has not. Early listening tests conducted in [1] showed that on average, humans are able to recognize everyday auditory contexts in 70% of cases, and confusions come among contexts that share the same prominent sound events. This study indicates that distinct sound events detected from auditory contexts are salient clues for human perception of audio context. As a result, an audio context can be characterized by the presence of individual sound events. For example, a railway recording might consist of "click-clacks" of a running train and crowds' "conversation", scattered with shout of "train horns". But not all the audio clips recorded on a train comply with the same event sequence, and this is why it underperforms if the annotation is done on long, complex and varied audio contexts. Consequently, in previous works, acoustic events in each training clip were manually labeled, which is both time consuming and labor intensive.

The motivation of this project is to recognize audio contexts by omitting the taxing workload of audio annotation. Rather than annotating along long and complex audio clips, massive singleton event-level audio files were needed, with corresponding audible concepts as their labels. To build a list of audible concepts, and to fill the gap between events and contexts, knowledge bases and descriptive text corpus came in handy.

A knowledge base is an information repository that provides a means for information to be collected, organized, shared, searched and utilized. In our case, we need to know what concepts are distinctively audible, that is, what concepts can make sounds or noise. Thus, the following taxonomic knowledge bases were adopted to give this information.

In this paper, we propose a sound context recognition framework using WordNet, Probase and several TV transcripts. Our approach assumes that different contexts, such as an office or a bar, can be distinguished by certain audible events. Consequently, contexts are modeled with an event set collected from TV transcripts. The proposed system can be divided into two phases, sound event detection and context recognition. A sound event detection phase is used to detect sound events that occur in a given clip. The detected event set is then compared with each context model to figure out where this clip was recorded. The system is evaluated with 120 testing clips, 10 for each context. The overall workflow of this framework is presented in Figure 1.1.

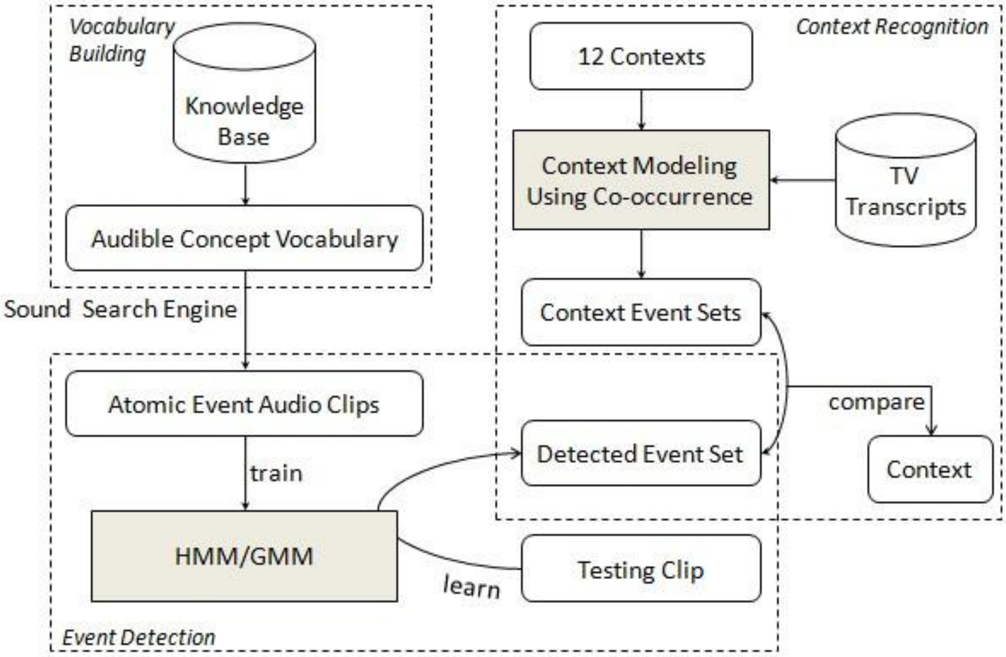


Figure 1.1 System Overview

The rest of this thesis is organized as follows. Chapter 2 gives a review of the related work, and Chapter 3 explains how the datasets were collected. Chapter 4 describes the context recognition system in detail. Experiments and discussions are presented in Chapter 5. Chapter 6 provides conclusions and suggestions for further study.

Chapter 2. Related Work

2.1 Environmental Sound Recognition

The research on context inference is known from many early works. However, most initial works made use of GPS location or a combination of sensor data collected by accelerometers, gyroscopes, thermometers and microphones. Early trials based on audio only has either been done by directly recognizing the context without explicitly detecting the individual sound events in the auditory scene, or simply focused on acoustic event detection without taking a step further into context inference.

Gaunard et al. [4] used HMMs to recognize five types of environmental noise events (car, truck, moped, aircraft, and train) using discrete HMMs. Best testing results came from a 5-state HMM using linear prediction cepstral coefficients (LPCCs) as features. A series of informal listening tests were conducted as a baseline, which showed that, on average, this system with 95.3% of correct classification outperforms human listeners who only achieves 91.8% for the same task.

El-Maleh et al. [5] classified five commonly encountered environmental noise contexts (car, street, babble, factory and bus) using a quadratic Gaussian classifier trained on a total of 18.75 minutes’ self-recorded audio clips. Their experiments showed that Linear Spectral Frequencies (LSF’s) are robust features in distinguishing different contexts. The system reached a precision of 80.5% in the testing stage, and contexts other than these five are unidentifiable.

In Peltonen’s work [6], the efficiency of different acoustic features and the effect of test sequence length were studied. They tried two classification systems: band-energy ratio features with 1-NN classifier and MFCCs with Gaussian mixture models. An accuracy of 70% for 25 different scenes was obtained on 30-second testing clips.

Eronen et al. [7] investigated the feasibility of an audio-based context recognition system, thus largely extended the number of recognizable everyday contexts to 24, such as a street, a restaurant, an amusement park, etc. A slight improvement in recognition accuracy was observed when linear data-driven transformations, i.e. Independent Component Analysis (ICA) or Linear Discriminant Analysis (LDA), were applied to low level audio features, like Zero Crossing Rate (ZRC), LPCC and MFCC. The authors also studied human listening tests, and found their proposed system performed rather well (69% recognition accuracy against 58% for the system and humans, respectively).

Till then, little research concentrated on the importance of significant sound events that happen in a recording. One of the approaches to use acoustic events in context inference is presented in [8]. Cai et al. proposed a flexible framework to recognize 5 auditory contexts (excitement, humor, pursuit, fight and air-attack) of a continuous stream, using a total of 10 predefined acoustic events - applause, cheer, laughter, etc. The optimal key event sequence is determined using Viterbi decoding, controlled by a 2-loop Grammar network defining probability of transitions between event pairs, according to the number of subsequent occurrences in training data. To infer the auditory context from the event sequence, a Bayesian network is adopted. A test on 12 hours of audio data gives a precision of 91.7% for event detection, and 82.4% for context inference. But due to the restriction on computational scale posed by Bayesian network, this framework performs well but only on limited datasets.

Lately, a more concise method of event-based context recognition is presented by Heittola et al. [9]. The method is based on representing each audio context using a histogram of audio events which are detected using a network of supervised

HMM/GMM classifier as explained in [10], where an accuracy of 24% was obtained in classifying isolated sound events into 61 classes (applause, bird, door, seatbelt, etc.). Individual sound events are detected in the unknown recording and a histogram of the sound event occurrences is built. Context recognition is performed by computing the cosine distance between this histogram and that of each context from the training database. An average classification accuracy of 89% is obtained in the recognition among 10 everyday contexts (beach, bus, hallway, etc.) using 61 types of audible events.

More recently, Lee et al. [11] came up with a novel approach with clip-level annotation. This refinement of the time axis represents a variant of Multiple-Instance Learning (MIL), which results in only frames that are most relevant to those labels modeled, whilst less informative frames are absorbed by the background models.

2.2 Audio Processing Using Knowledge Base

Besides paying attention to acoustic event detection in recognizing sound contexts, another means to solve vast audio data is to introduce knowledge bases. In sound analysis works, no matter speech analysis or sound computing, in order to prepare training set, researchers need to access vast collections of audio clips and manually annotate their content one by one. The taxonomy and various relations that knowledge bases provide make it possible to alleviate some manual workload.

WordNet ([2], Miller, 1995: 39-41) is a lexical database of English containing nouns, verbs, adjectives and adverbs, and organizes senses of concepts in synonym sets (synsets), with links between the senses of relationships like: broader/narrower sense (is-a relation), part-of relation, made-of relation and so on. For instance, it knows that “cry" is a “sound”, and the word “drawer” has 2 senses, which means it refers to either “a boxlike container in a piece of furniture”, or “an artist skilled at drawing”.

Figure 2.1 displays part of the WordNet structure, with each rectangle representing a synset, and each line segment indicating a is-a relation.

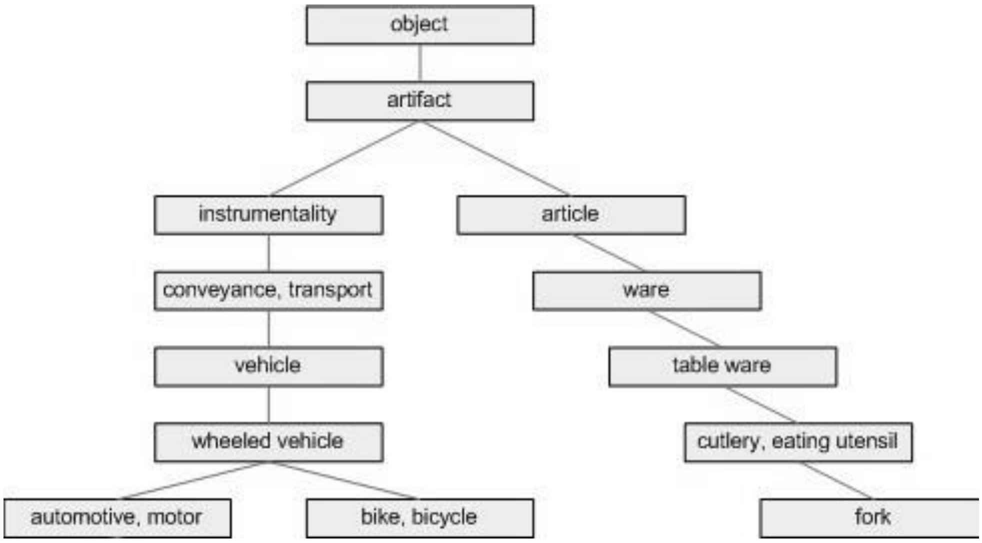
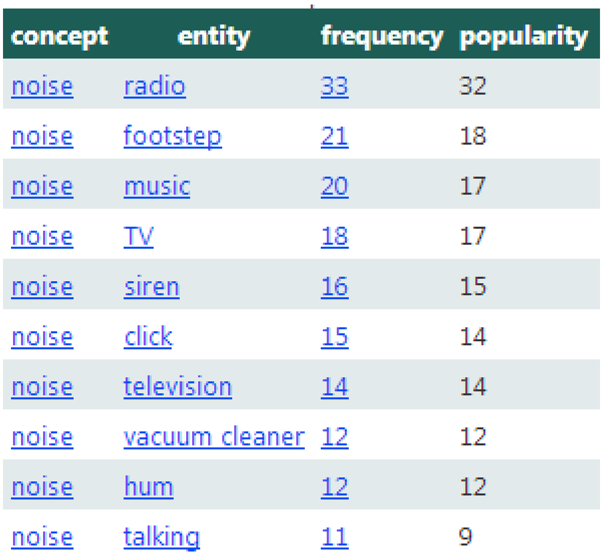


Figure 2.1 Taxonomy in WordNet

Probase ([3], Wang et al., 2012: 481-492) is a probabilistic taxonomy. The core taxonomy of Probase (Probase v5.2) contains over 2.7 million concepts, connected by hypernym/hyponym (concept/instance) relations. For each concept-instance pair, e.g. “company” and “apple”, Frequency (I=apple | C=company) is provided, depicting how frequently people refers to the Apple Inc. when they mention a company. Figure 2.2 shows the top 10 hyponyms of “noise” found in Probase.



concept	entity	frequency	popularity
noise	radio	33	32
noise	footstep	21	18
noise	music	20	17
noise	TV	18	17
noise	siren	16	15
noise	click	15	14
noise	television	14	14
noise	vacuum cleaner	12	12
noise	hum	12	12
noise	talking	11	9

Figure 2.2 Top 10 Noise in Probase

Cano and Koppenberger [12] proposed a solution to make manual annotation easier. Sound samples were gathered and tagged with unambiguous glosses (descriptions of each sense) in WordNet. A 20-nearest-neighbor classifier was trained to annotate more audio samples using normalized Manhattan distance. Based on 15 sound effects (truck, aircraft, moped, etc.), an annotation test on 261 audio files showed an accuracy of 91%.

Based on this trial, the authors further built sound effect taxonomy [13] and processed audio retrievals [14] with the assistance of WordNet. In [13], 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. Afterwards, in [14], the researchers presented a sound effect retrieval system that incorporates content-based audio techniques and semantic knowledge provided by WordNet.

Chapter 3. Data Collection

3.1 Audio Clip Crawling

Audio clips were downloaded according to an audible concept vocabulary built in Section 4.1 using WordNet and Probase. For each atomic event in our vocabulary, we queried them on 3 sound search engines, SoundJax[1], FindSounds[2], and Freesound[3]. However, due to the flexibility of search engines' matching method and ranking strategy, we did not download all the audio clips returned by the sound search engines. Instead, we designed a set of rules to determine whether or not we should fetch a clip.

Firstly, we examined whether the clip played the same event as its query word. Audible concepts in our vocabulary can be generally divided into two categories, single-word concepts and multi-word concepts. For each single-word concept, we built a synset for it containing itself, its synonyms and all its inflected forms in WordNet. Then we checked the filename and labels of the returned clip to see whether they were in the synset. If yes, this clip became a candidate clip to download. As for multi-word concepts, a similar examination was conducted on each word component. Only when all the word components or their synonyms appeared in the filename or label can this clip pass this examination stage.

For clips that matched the same theme as we queried, we would check their duration. This was based on the assumption that a clip that is too long in duration is very likely to contain more than a single event. Since we were to label the clips automatically with the query word, this impurity will definitely cumber with the recognition

[1] http://soundjax.com
[2] http://www.findsounds.com/
[3] http://www.freesound.org/

performance. Hence, after checking out 100 clips with various durations, we restricted that the duration of qualified clips should be less than 40 seconds.

Finally, in the light of these filtering rules, we crawled a total of 145 thousand clips for 2604 events. On average, there were 55 clips for each event, which are sufficient to train a robust HMM classifier.

3.2 TV Transcripts Collection

Compared with previous work, we no longer have long training clips with a sequence of events. As a consequence, we needed another dataset to bridge this gap between event set and context. This dataset was supposed to provide a rich set of information on what events happen in any context. In this aspect, TV transcripts have two advantages. It not only describes what happens in a context, but also presents this information in a good manner. In well-written transcripts, whenever the plot switches to a new place, there is a special sentence in a certain pattern indicating where the characters are now, such as “Scene: Central Perk” or “CUT TO: Sheldon’s apartment”. This makes it easy to extract the context from a short sentence, and can help us position the range to collect events for this context accurately.

At this stage, a challenge was that there were little websites that gather and formalize transcripts officially. So we could only download transcripts via online searching. The procedure is hence very time consuming, and the format of transcripts differs, which results in further manual formalization among transcripts. We downloaded as many transcripts as we could, referring to the Most Voted TV Series from Internet Movie Database (IMDb[4]). Among the top 50 voted series, we downloaded transcripts of 30 series. Their name, genre, publication year and the number of available episodes are shown in Table 3.1. For the last column, green denotes the transcripts were fully

[4] http://www.imdb.com/search/title?sort=num_votes,desc&title_type=tv_series

downloaded, whilst yellow means only part of the transcripts were found.

Table 3.1 A Log for Downloaded Transcripts

Series Name	Genre	Year	Statistics
The Big Bang Theory	Comedy	2007	126 episodes (6 seasons)
Friends	Comedy, Romance	1994	229 episodes (10 seasons)
How I Met Your Mother	Comedy, Romance	2005	135 episodes (6 seasons)
Prison Break	Crime, Mystery, Thriller	2005	23 episodes (2 seasons)
Lost	Adventure, Fantasy, Mystery, Sci-Fi, Thriller	2004	118 episodes (6 seasons)
Sherlock	Crime, Mystery	2010	6 episodes (2 seasons)
Family Guy	Animation, Comedy	1999	104 episodes (7 seasons)
South Park	Animation, Comedy	1997	232 episodes (16 seasons)
Arrested Development	Comedy	2003	22 episodes (1 season)
Scrubs	Comedy	2001	150 episodes (7 seasons)
Modern Family	Comedy	2009	84 episodes (4 seasons)
House M.D.	Mystery	2004	177 episodes (8 seasons)
Supernatural	Fantasy, Horror, Mystery, Thriller	2005	167 episodes (8 seasons)
The Vampire Diaries	Fantasy, Horror, Mystery, Romance, Thriller	2009	82 episodes (4 seasons)
Firefly	Adventure, Sci-Fi	2002	11 episodes (1 season)
True Blood	Fantasy, Mystery, Romance, Thriller	2008	34 episodes (5 seasons)
Seinfeld	Comedy	1990	179 episodes (9 seasons)
Fringe	Mystery, Sci-Fi, Thriller	2008	86 episodes (4 seasons)
Glee	Comedy, Musical, Romance	2009	27 episodes (3 seasons)
The Simpsons	Animation, Comedy, Family	1989	17 episodes (11 seasons)
Futurama	Animation, Comedy, Sci-Fi	1999	117 episodes (7 seasons)
Battlestar Galactica	Action, Adventure, Sci-Fi	2004	63 episodes (4 seasons)
The X Files	Mystery, Sci-Fi, Thriller	1993	160 episodes (7 seasons)
Once Upon a Time	Adventure, Fantasy, Mystery, Romance	2011	38 episodes (2 seasons)
That '70s Show	Comedy, Romance	1998	91 episodes (6 seasons)
Buffy the Vampire Slayer	Action, Fantasy	1997	143 episodes (7 seasons)
Desperate Housewives	Comedy, Romance	2004	78 episodes (6 seasons)
Doctor Who	Adventure, Family, Sci-Fi	2005	249 episodes (7 seasons)
Smallville	Adventure, Romance, Sci-Fi	2001	103 episodes (8 episodes)
Bones	Comedy, Crime, Mystery, Romance	2005	112 episodes (6 seasons)

3.3 Testing Clips Collection

To evaluate the performance of our system on 10 daily contexts (bar, beach, cafeteria, church, concert, office, park, street, toilet and train), we downloaded 100 testing clips from YouTube[5], FindSounds and Freesound. For each context, 10 audio clips were crawled.

There were also selection criteria. Firstly, the clip should be no longer than 4 minutes. In fact, 79% of testing clips we downloaded are within 1 minute. Secondly, we checked the acoustic fidelity to ensure that most of the audio clips are relatively recognizable by human-beings. At last, we cropped 13 clips because the context transformed from one to another during the clip, which exceeded our topic.

[5] http://www.youtube.com

Chapter 4. System Implementation

The system aims to carry out context inference after detecting acoustic events in a given audio clip. As to training data, unlike in [5] [8] and [9], we collected quantities of short audio files each containing a singleton sound event, rather than use a few long environmental sound recordings. Short audios were obtained through sound search engines, and labeled with its query word instead of by manual annotation. Considering which audio files meet our requirement, we defined a vocabulary of audible concepts using WordNet and Probase in advance. With a mass of short audio clips, features were extracted and HMMs are trained for each event. Similar to representing contexts as histograms of acoustic events in [9], a co-occurrence matrix between contexts and events was constructed using TV series transcripts. In this manner, contexts are modeled as a vector of events. When a sequence of events during a testing clip are detected, the system will compare this event set with those of predefined contexts, afterwards estimate the most probable contexts by ranking the contexts according to the number of shared events.

4.1 Build an Audible Concept Vocabulary

4.1.1 Initiate the Audible Concept Vocabulary

This audible concept vocabulary defines what events we are concerned about. So we tried our best to make it complete and strict. In the first place, we initiate the audible concept vocabulary with the sound types listed in FindSounds[6], MediaCollege[7] and Soundrangers[8], instances of “sound”, “noise”, “sound effect” and “animal” in Probase

[6] http://www.findsounds.com/types.html
[7] http://www.mediacollege.com/downloads/sound-effects/
[8] http://www.soundrangers.com//index.cfm?category=1&left_cat=1

(over 5500 instances, e.g. bell, rain, radio, footstep, dog, cat etc.), and hyponyms of “sound” and “noise” in WordNET (over 200 concepts, e.g. ring, beat, bell, chatter, crack etc.).

During the initialization procedure, we found many concepts extracted from Probase incorrect. In a subjective check over a sample size of 100 concepts, an error rate of 10% (e.g. l.p international, ghostbusters musical theme) was obtained and 38% were names of particular animal species (small clawed otter, dofassa waterbuck), which are of little use. Thinking of the form inconsistency of some Probase concepts and the frequency information it provides, frequency filtering were applied after word form inflection. The filter approach on Probase can be described in following steps:

1. For one-word concepts like “cars” and “plane”, the singular and plural format are merged, which increases their frequency.

2. Multi-word concepts with improper punctuations (e.g. majority of “spirit they've vanished”) were abandoned.

3. Some concepts were too specific and can be replaced by their hypernyms. For instance, “swamp deer”, “musk deer” and “red deer” are all instances of “deer”. For such concepts with more than one word, examine whether each of its composing word already exists in our vocabulary. If yes, just yield its frequency to that one-word concept, and delete it. This partly avoids improperly specific concepts.

4. With reference to WordNet synsets, we merged synonyms to enhance their frequency so that they will not be discarded during the subsequent frequency filtering. For example, “plane” (Frequency (I=plane | C=sound) = 3) and “airplane” (Frequency (I=airplane | C=sound) = 3) contribute a frequency of 3 to each other, as a result, both of them were to survive the filtering.

5. Till then, we could filter all the Probase concepts with a threshold of frequency>5. This succeeded in throwing away the long tail of weird or indistinctive concepts.

After these 5 steps, the number of highly-confident Probase audible concepts reduces from over 5000 to 651.

4.1.2 Concept Lemmatization

When we merged all the aforementioned seeds together, we got an initial concept vocabulary of size 1594. Now, the importance of concept representation emerges. A good representation ensures high precision and recall in subsequent audio clip retrieval as well as event-context co-occurrence collection. Observed from our initial vocabulary, audible concepts were no more than sound makers (“helicopter”, “seagull”, etc.), predicates (“drop”, “peel out”, etc.), sometimes with either subjects (“children giggle”, “dog bark”, etc.) or objects (“shoot a gun”, “shake a can”, etc.) as its arguments, or onomatopoeia refined by its scenario or sound source (Whoosh Sounds: Fire, Whoosh Sounds: Rocket). A concept integrator was needed so that given a short audible phrase, a lemmatized representation will be returned. We lemmatize the concepts as follows:

1. We used WordNet stemmer to stem words. This procedure cast the words into lower case and singular form. If WordNet does not contain the word, only cast it into lower case.

2. Discard some redundant components from the concept, like “… sfx”, “… noise”, “… sound effects”, “… ambience”, “sound of…”. For instance, cast “button click sound effects” into “button click”, and “sweeping noise” into “sweeping”.

3. Cast present participles and past participles into the base form of that verb.

The recognition of present participles can not rely on WordNet stemmer. So four more rules were designed:

1. A present participle should be a verb in WordNet and ends with suffix “–ing”.

2. If it is also an adjective in WordNet, such as “interesting”, it is not a present participle.

3. If the number of its noun senses is smaller than one third of that of its verb senses, it is a present participle.

4. Otherwise, see the glosses of its noun senses. If these glosses contain some expressions that indicate a present participle, such as "… action of…" and "… process of...", it is a present participle.

4.2 Expand the Vocabulary

So far, the vocabulary is clean enough. But its size is a bit small. Thus, an expansion was tried out. The expansion was done iteratively. Each iteration could be divided into two phases, SSE filtering and expansion using Probase and WordNet. The filtering is necessary because it ensures that we give a pure seed for the upcoming boosting. At the meantime, SSEs also helps enlarge our vocabulary by adding in relevant concepts.

4.2.1 Filter By SSEs

In order to guarantee the correctness of the vocabulary, we came up with an online verification method using sound search engines. The steps are shown below:

1. Query every audible concept on FindSounds, SoundJax and Freesound.
2. Exclude files whose duration is longer than 40 seconds.
3. Exact file name match and synonym match are both conducted.
4. Tally frequent filenames or tags that are not yet in our vocabulary, e.g. we added "alarm bell" in when querying "door bell".
5. If scarcely any filename exactly matches the query, figure out what components are taken as the query word in this query. Record their combination as a new concept.

4.2.2 Expansion Using Probase and WordNet

After filtering, what we had was considered reliable for expansion. We then bootstrapped more audible concepts by adding in confident siblings of existing concepts. Figure 4.1 illustrates how we look into WordNet and determine whether or not to add a sibling. An expansion using Probase follows the same manner. On the left lies our vocabulary to expand. We look up hypernyms of every concept that we have now, and find that “alarm bell”, “siren alarm” and “car horn” share the same hypernym, “alarm system”. Then attention is paid to hyponyms of “alarm system” that are not yet in the vocabulary, such as “burglar alarm”, “smoke alarm” and “fire alarm”. So in this iteration, we will add them into our vocabulary.

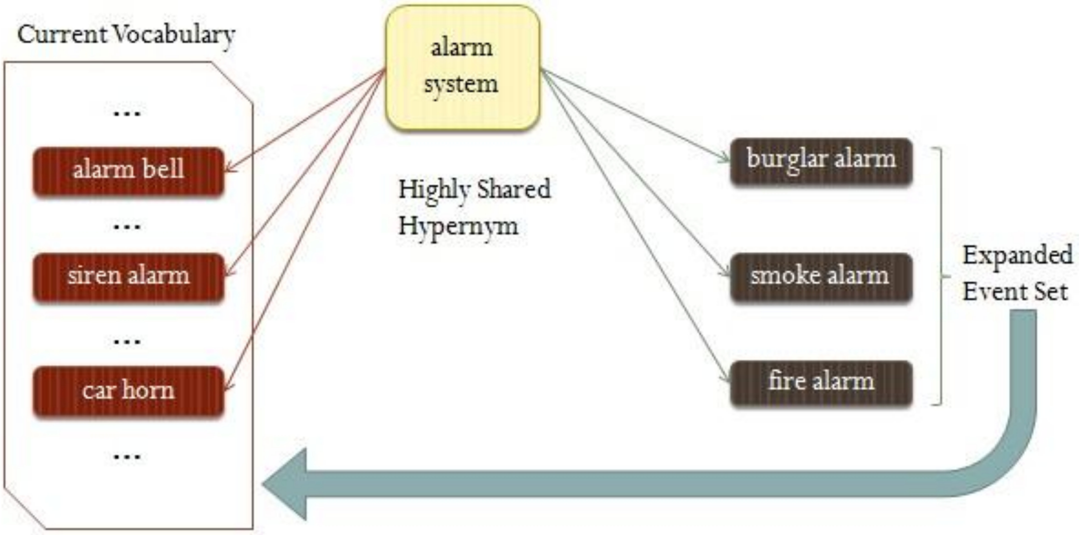


Figure 4.1 An Example of Vocabulary Expansion Using WordNet

Of course, nearly 2000 concepts in our vocabulary have quantities of hypernyms, which further result in even more candidate hyponyms to add in. Among these candidates, a large proportion is noise. For instance, a “phone” is a typical acoustic event, and has a hypernym, “electronic equipment”. A “modem” is also a piece of “electronic equipment”, but is not very audible. So it is quite out of our concern. In order to avoid such noise, two thresholds were designed in formula 4-1 and 4-2, where h is a hypernym and V denotes our present vocabulary.

$$s(h,V) = \#\ of\ hyponyms\ of\ h\ that\ exist\ in\ V \tag{4-1}$$

$$\begin{aligned} p(h,V) &= proportion\ of\ covered\ hyponym \\ &= \frac{s(h,V)}{the\ total\ \#\ of\ hyponyms\ oj} \end{aligned} \tag{4-2}$$

Here, s(h,V) intuitively shows how much h is related to V. And p(h,V) avoids considering too-general hypernyms, say “entity”, because it is an uninformative hypernym but highly shared by concepts in V.

In practice, we did not manually set the value of threshold s and p. We automatically tuned it by experiments. Different combinations of s (from 3 to 40, inclusive) and p (from 0.3 to 0.9, inclusive) were tried. After each expansion trial, results were examined by SSEs as described in Section 4.2.1. Then a score was calculated according to formula 4-3, where true or false is the examination result returned by SSEs. As this score defined, we want to not only include more true events but also ensure the expansion accuracy to be relatively high.

$$Escore = log10(tn) * (tn/(fn+1))$$

$$, where\ tn\ is\ \text{the}\ \#\ of\ true\ events, fn\ is\ the\ \#\ of\ false\ events \qquad (4\text{-}3)$$

After experiments, we determined the expansion (s, p) pair for WordNet is (4, 0.4), and for Probase it is (32, 0.5). After 8 iterations, the vocabulary enlarged from 1594 concepts to 4603 concepts.

4.3 Download Short Audio Files and Automatic Annotation

As described in Section 3.1, we queried audio files for each audible concept from SoundJax, FindSounds, and Freesound. Each audio file was labeled with its query word.

4.4 Co-Occurrence Collection

Using a similar method as described in Section 4.1.1, a sound context vocabulary was built using WordNet. We collected hyponyms of “scene”, “occasion”, “social event”, “construction”, “facility”, “vehicle”, “organization”, “field”, “geological formation”, "geological phenomenon", "building", "public transport", "high-traffic area", "workplace", "eating place", "group action", "road", "house", "way", "room", "terminal", "place of business", "landsite", "city district", "rural area", "land", "social group", "geographical point" and "public holiday" in WordNet (e.g. "farmland", "bakery", "café", etc.).

Now, we had two vocabularies defining all the contexts and concepts we are concerned about. It was time to collect the co-occurrence information between events and contexts, that is, to see what events can happen in a particular context. This information was extracted from TV transcripts we crawled in Section 3.2.

In the first place, we should be aware of the context of every scene. As aforesaid, scene indicators are usually distinctive and short sentences. The context extraction is clearly shown below:

1. Apply exact match to find out contexts that appear in a scene indicator.
2. Adopt Named Entity Recognition (NER) technique to find all the location names, organization names as well as person names. This step and the subsequent two aim to solve problems that cafeterias like "Cheesecake Factory" and "Central Perk" are not recognized.
3. For each location/organization/person name recognized in step 2, look it up in Probase taxonomy to see whether it belongs to a category that exists in our context vocabulary, such as in the previous example, "Cheesecake Factory" and "Central Perk" are both instances of "cafeteria".
4. When a named entity is affirmed to be a context, check whether some component of it was wrongly considered as a context. If yes, remove it. For instance, during the exact match, the system once thought "factory" is the context indicated by sentence "Scene: a Cheesecake Factory in a street corner".

Once we can point out where the paragraphs between two scene indicators took place, we can collect as many events that co-occurred with this context. This was done obeying three rules are as follows:

1. Exact match works for one-word event. Split the words within a multi-word event, and check their existence within a sentence one by one, neglecting the sequence they appear.

2. Literally, it is hard for machines to understand whether the word “can” is a modal verb or refers to a metal container. POS tagging is a great helper in this respect.

3. Along with event collection, the significance or importance of an event for a specific context was assigned using the SigScore(e) defined in formula 4-4. The definition to SigScore emphasizes three aspects for a significant event. The more it co-occurs with a context, the more relevant they seem to be. But some events are too prevalent to distinguish among all the contexts where they appear, such as “people” and “conversation” are everywhere. The above two aspects are the same as what term frequency-inversed document frequency (tf-idf), a classical method in information retrieval, concerns about. One more thing we focused on was the audibility of an event, proportional to the number of clips for it. For instance, in our real life, we can rarely hear the sound of eating cake but the sound of cutlery are heard in almost every cafeteria. Events that co-occurred with a context are ranked from high to low according to this SigScore.

$$\begin{aligned} SigScore(e) &= frequency\ of\ e\ in\ corpus \\ &\quad * log(\#\ of\ contexts/\#\ of\ contexts\ that\ e\ co-occurred\ with) \\ &\quad * log(\#\ of\ clips) \end{aligned} \tag{4-4}$$

For each context, we only reserved the top 100 events. To make our further evaluation more feasible, we selected 10 contexts, that is, office, park, street, cafeteria, store, bar, concert, church, toilet and beach. They are the most frequent as well as most dissimilar contexts in terms of the number of shared events. Then, we manually filtered some events that seemed to be topics discussed in each context rather than things actually happened. As a consequent, every context was modeled as a set of remaining events.

4.5 Feature Extraction

Temporal features and frequency features describes a piece of audio informatively in a mathematic way. According to a survey conducted in [15] and experiments on feature selection in [7] [8] [16], combining MFCCs, their first- and second-order differentials, LPCCs and ZRCs can boost the recognition performance. We extracted the most promising feature for HMM training, i.e. 13-dimension MFCC with its 1st- and 2nd-differentials.

In sound processing, the mel-frequency cepstrum (MFC) is a frequency domain feature. It indicates the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. MFCCs are coefficients that collectively make up an MFC. It is the most commonly used feature in speech recognition, music genre classification and environmental sound recognition (ESR).

This step and the next were accomplished using an open source toolkit HTK [18], providing sophisticated facilities for audio annotation, audible feature analysis, HMM training, testing and results analysis. The software supports HMMs using both continuous density mixture Gaussians and discrete distributions and can be used to build complex HMM systems.

4.6 Acoustic Event Detection

An HMM is a statistical Markov model where the system being modeled is assumed to be a Markov process with unobservable states. Other than in a simple Markov chain, in which the state is visible, in a hidden Markov model, the state is not directly visible, but output, dependent on the state, is visible. Each state has a probability distribution over a set of possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of latent states. HMMs are especially known for applications in temporal pattern recognition such as speech, music information retrieval (MIR), handwriting, part-of-speech (POS) tagging, and bioinformatics.

According to experiments on the number of states and the topology of HMM in [7] [8] [10] [17], a number of 5-state left-to-right HMMs were trained with the Baum-Welch training procedure, one for each acoustic event. The probability density of each state was modeled using GMM having 16 components. “Left-to-right” restricts the transition logic of an HMM, meaning that once the state jumps, it will never go back. Figure 4.2 gives a clear view of the topology of a 3-state left-to-right HMM.

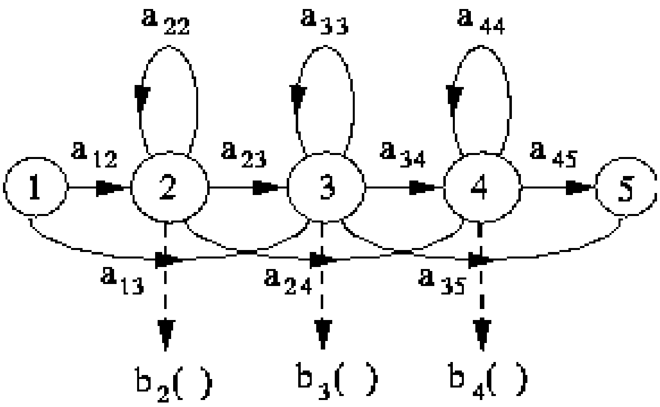


Figure 4.2 A 3-State Left-to-Right HMM

In the testing stage, event detection over the entire recording was done using the Viterbi algorithm to obtain the most likely event sequence. A file written in context-free grammar defines the network topology of singleton HMMs. It assigns transition probability equally to event pairs that co-occurred within a same context, as collected in Section 4.4, while zero this probability between those never appeared together in TV transcripts.

4.7 Sound Context Inference

Through event detection in Section 4.6, the acoustic event vector v of an unknown recording was detected. A comparison between v and each event set s_i of context c_i (i from 1 to 10) was done by simply accumulating the shared event count. The reason why we did not adopt cosine similarity was that cosine similarity gives a penalty to contexts with more co-occurred event information. This unfairness turned out to largely reduce the recognition accuracy for contexts like office (33 co-occurred events collected) and cafeteria (28 co-occurred events collected).

Chapter 5. Evaluation

As mentioned in Section 3.3, the testing audio clips are collected from YouTube, FindSounds and Freesound. This chapter will firstly present the testing result of our system in general. Then we will go into details to see why for some particular contexts, our system performs not satisfying enough.

5.1 Experiments

We are evaluating the performance of our system on 10 daily contexts, that is, bar, beach, cafeteria, church, concert, office, park, street, toilet and train. For each context, 10 audio clips were prepared. Table 5.1 and Figure 5.1 both illustrated the recognition accuracy for different contexts, where “top1” and “top3” refers to the number of most likely contexts taken into consideration. An average accuracy of 56% was obtained in this experiment.

Table 5.1 Context-wise Recognition Accuracy for 10 Contexts

	bar	**beach**	**cafeteria**	**church**	**concert**	**office**	**park**	**street**	**toilet**	**train**
top1	10%	50%	10%	40%	10%	40%	20%	80%	50%	20%
top3	20%	50%	80%	40%	30%	60%	60%	90%	60%	70%

Table 5.2 Correlation Between Recognition Performance And # of Events

	bar	**beach**	**cafeteria**	**church**	**concert**	**office**	**park**	**street**	**toilet**	**train**
# of events	17	8	28	9	12	33	20	19	13	10
# of correct recognition	2	5	8	4	3	6	6	9	6	7

Table 5.2 looks into it in more details by trying to find out whether recognition performance is related to the number of events that co-occurred with each context in

our TV transcripts. As can be seen intuitively from Figure 5.2, as the number of co-occurred events increases or decreases, as indicated by the blue line, the recognition accuracy rises or declines accordingly in most of the time. A lower bound of 18 co-occurred events almost ensures a high accuracy (72.5%) in recognizing among these 10 contexts.

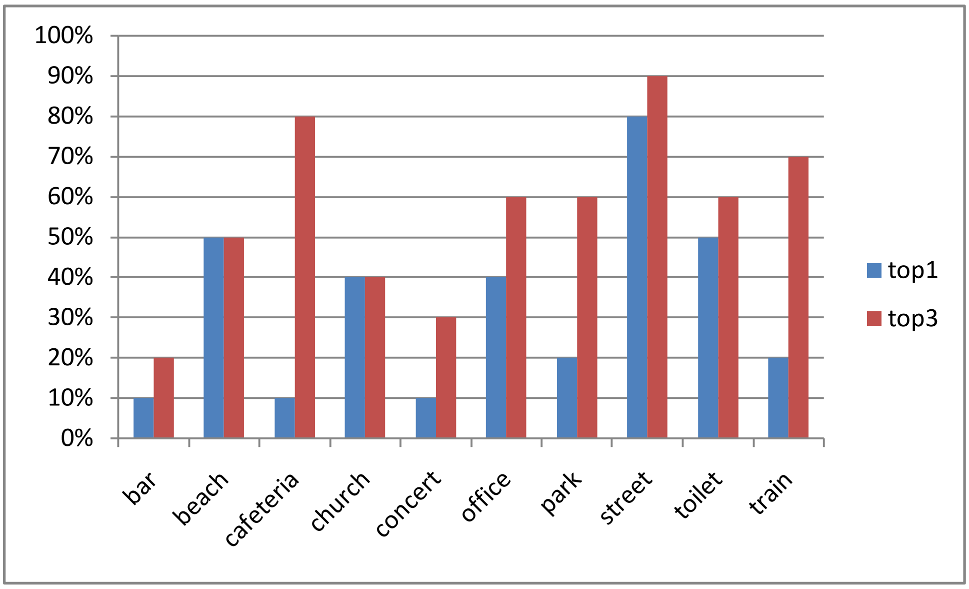


Figure 5.1 Bar Chart for Context-wise Recognition Performance

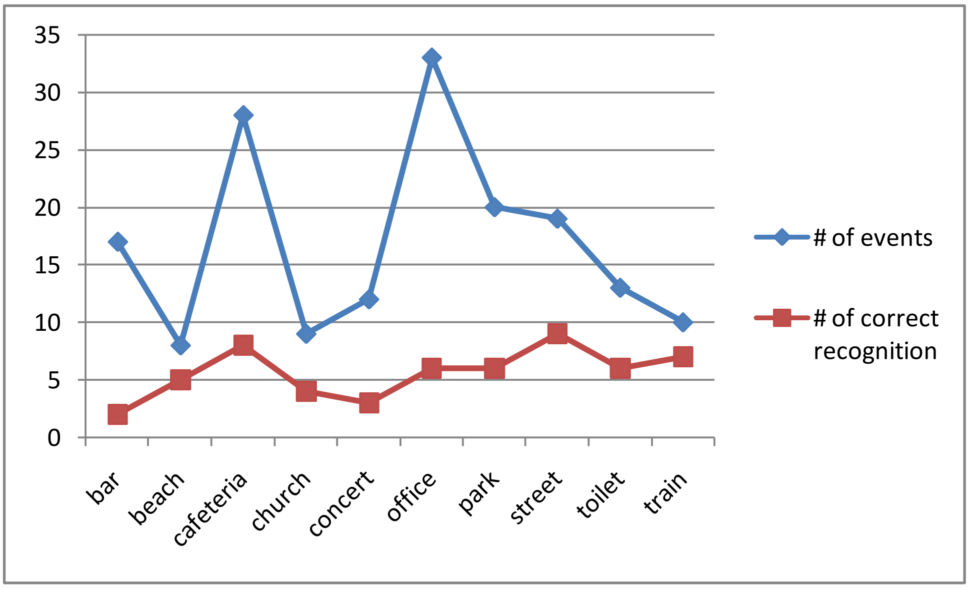


Figure 5.2 Correlation Between Recognition Performance And # of Events

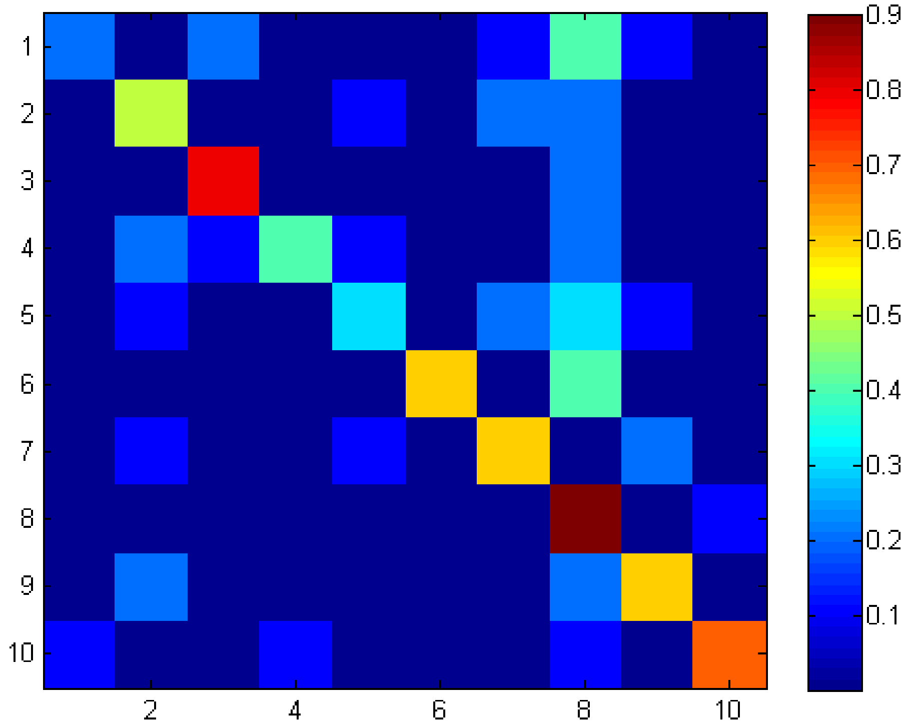
5.2 Causal Analysis

The confusion matrix in Table 5.3 and plotted in Figure 5.3 provides us an entry to investigate the reason for the poor recognition performance on some particular contexts, such as bar, beach, church and concert. The column on the left denotes the real context, whilst the first row refers to the context recognized by our system.

Table 5.3 Confusion Matrix for 10 Context Recognition

	bar	beach	cafeteria	church	concert	office	park	street	toilet	train
bar	2	0	2	0	0	0	1	4	1	0
beach	0	5	0	0	1	0	2	2	0	0
cafeteria	0	0	8	0	0	0	1	2	0	0
church	0	2	1	4	1	0	0	2	0	0
concert	0	1	0	0	3	0	2	3	1	0
office	0	0	0	0	0	6	0	4	0	0
park	0	1	0	0	1	0	6	0	2	0
street	0	1	0	0	0	0	0	9	0	1
toilet	0	2	0	0	0	0	1	2	6	0
train	1	0	0	1	0	0	0	1	0	7

Figure 5.3 Confusion Matrix for 10 Context Recognition



For the context “bar”, 40% of testing clips are thought to be recorded in a street according to our system. The reason was that we did not find drum set as a typical bar event. So the noisy sound it made was mostly recognized as car starting, which further led to the context “street”. Besides, the scene in a bar is in essence very similar to that inside a cafeteria, so the distraction onto “cafeteria” is tolerable.

As for a church, it seemed quite strange that some sharp church tolls are considered as car brakes. Apart from that, some resonant and rhythmic chants recorded from a certain distance sound like ocean waves lapping the coast. That’s the main reason for recognizing a “church” as a “street” or “beach”. Similarly, the sound of a copier, a fax or a printer in an “office” was sometimes mistaken for a car engine in a “street”.

When it comes to the “concert” scene, some high pitch durations of wind instrument were mistaken as car sirens. Consequently, it was sometimes thought to be in a “street”. Piano is a significant event for both concert and church, thus piano alone can not distinguish the two contexts.

Moreover, it can be easily found that there was not low an error rate among “cafeteria”, “toilet” and “beach”. That is caused by the similarity on hearing when the sound of pouring drink, toilet flush and ocean wave are recorded in different distances. A same problem exists among “cafeteria”, “street” and “office” because the sound of coffee maker, bus engine and copier are not very distinguishable.

Chapter 6. Conclusion

In daily subjective context recognition, a highly recognizable event that only happens in some particular scenarios plays an irreplaceable role. In recent years, works on environmental sound recognition pay increasingly more attention to improve the performance of event detection as well as optimize context inference with the event sequence detected.

In this thesis, we concentrated far more than others on events. In the training stage, we even called off context-level training clips and the most prudent part of previous work, manual annotation. We introduced the help of text into sound processing, both taxonomical knowledge bases and descriptive text corpus. Even without any complement or alignment from any long and complicated real life recordings, we tackled the context recognition problem with a competitive accuracy of 56%.

The novel part occupied most our time in implementing the system, while some classical ESR stages need further improvements. To sum up, we also summarize some future tasks to hopefully optimize this system:

1. Enlarge our text corpus to let in more co-occurrence information and cripple the relative frequency of topic events.

2. Do event detection with a combination of multiple audio features, especially some temporal features like ZRC.

3. Customize the number of GMM components for different events, so that the system neither over fit some sound samples nor miss any uncommon sounds.

4. Try to replace the co-occurrence collection session by some other techniques, such as Google similarity distance defined in [19] or relational knowledge base, to find out possible events that happen in a particular context.

REFERENCE

[1] Peltonen, Vesa T K, Antti J. Eronen, et al. Recognition of everyday auditory scenes: potentials, latencies and cues [J]. In Proc. 110th Audio Eng. Soc. Convention, 2001.

[2] Miller, George A. WordNet: a lexical database for English [J]. Communications of the ACM, 1995, 38(11): 39-41.

[3] Wu Wentao, Hongsong Li, Haixun Wang, et al. Probase: A probabilistic taxonomy for text understanding [C]. Proceedings of the 2012 international conference on Management of Data, 2012, 481-492.

[4] Gaunard Paul, Mubikangiey C G, Couvreur C, et al. Automatic classification of environmental noise events by hidden markov models [C], IEEE International Conference on Acoustics, Speech and Signal Processing, 1998, 6: 3609-3612.

[5] El-Malch K, Samouclian A, Kabal P. Frame Level Noise Classification in mobile environments [C], IEEE International Conference on Acoustics, Speech and Signal Processing, 1999, 1: 237-240.

[6] Peltonen V T, Tuomi J T, Klapuri A, et al. Computational auditory scene recognition [C], International Conference on Acoustics, Speech and Signal Processing, 2002, 2: 1941-1944.

[7] Eronen A J, Peltonen V T, Tuomi J T, et al. Audio-based context recognition [J]. IEEE Transactions on Audio, Speech, and Language Processing, 2006, 14(1): 321-329. Mesaros A, Heittola T, Eronen A, et al. Acoustic event detection in real life recordings [C]. 18th European Signal Processing Conference, 2010, 1267-1271.

[8] Cai Rui, Hanjalic A, Zhang Hongjiang, et al. A flexible framework for key audio effects detection and auditory context inference [J]. IEEE Transactions on Audio, Speech, and Language Processing, 2006, 14(3): 1026-1039.

[9] Heittola T, Mesaros A, Eronen A, et al. Audio context recognition using audio

event histograms [C]. Proc. of the 18th European Signal Processing Conference, 2010, 1272-1276.

[10] Mesaros A, Heittola T, Eronen A, et al. Acoustic event detection in real life recordings [C]. 18th European Signal Processing Conference, 2010, 1267-1271.

[11] Lee K, Ellis D PW, Loui A C. Detecting local semantic concepts in environmental sounds using markov model based clustering [C]. International Conference on Acoustics, Speech and Signal Processing, 2010, 1: 2278 - 2281.

[12] Cano P, Koppenberger M. Automatic Sound Annotation [C]. 14th IEEE Workshop on Machine Learning for Signal Processing, 2004, 391-400.

[13] Cano P, Koppenberger M, Herrera P, et al. Sound effects taxonomy management in production environments [C]. Proc. AES 25th Int. Conf, 2004.

[14] Cano P, Koppenberger M, Groux S L, et al. Knowledge and content-based audio retrieval using WordNet [C]. Proc. of the International Conference on E-business and Telecommunication Networks, 2004, 301-308.

[15] Mitrović D, Zeppelzauer M, Breiteneder C. Features for content-based audio retrieval [J]. Advances in Computers: Improving the Web, 2010, 78: 71-150.

[16] Zeng Zhi, Li Xin, Ma Xiaohong, et al. Adaptive context recognition based on audio signal [C]. 19th International Conference on Pattern Recognition, 2008, 1-4.

[17] Reyes-Gomez M J, Ellis D PW. Selection, parameter estimation, and discriminative training of hidden markov models for general audio modeling [C]. 2003 International Conference on Multimedia and Expo, 2003, 1: 73-76.

[18] Young Steve J, Young Sj. The HTK hidden Markov model toolkit: Design and philosophy [M]. University of Cambridge, Department of Engineering, 1993.

[19] Cilibrasi R L, Vitanyi Paul MB. The google similarity distance [J]. IEEE Transactions on Knowledge and Data Engineering , 2007, 19(3): 370-383.

ACKNOWLEDGEMENTS

I owe a debt of gratitude to Professor Kenny Zhu, my advisor, for the vision and foresight that encouraged me a lot throughout this project. The conception of this cross-domain (i.e. knowledge base and sound computing) trial was fully inspired by him. And all along my trip to fulfill this task, he gave me plenty of wise suggestions, most of which has enlightened me to pass through great challenges. It was his full support, strong sense of responsibility and concern that brought our project to a successful end, indeed in selfless spirit.

It is also my duty to record my thankfulness to Dr. Ye Wang, an associate professor from the School of Computing, National University of Singapore, and Dr. Xinxi Wang, an excellent student of Dr. Ye Wang. Many thanks for their generous advice in sound processing, which is an important component of my work.

Finally, I take this opportunity to acknowledge the favor of the total team of Prof. Zhu and Dr. Wang who collaborated and helped in producing this work.

With warm regards,
Menglu Li