On the Applicability of Speaker Diarization to Audio Indexing of Non-Speech and Mixed Non-Speech/Speech Video Soundtracks

Abstract: A video‘s soundtrack is usually highly correlated to its content. Hence, audio-based techniques have recently emerged as a means for video concept detection complementary to visual analysis. Most state-of-the-art approaches rely on manual definition of predefined sound concepts such as “engine sounds”, “outdoor/indoor sounds”. These approaches come with three major drawbacks: manual definitions do not scale as they are highly domain-dependent, manual definitions are highly subjective with respect to annotators and a large part of the audio content is omitted since the predefined concepts are usually found only in a fraction of the soundtrack. This paper explores how unsupervised audio segmentation systems like speaker diarization can be adapted to automatically identify low-level sound concepts similar to annotator defined concepts and how these concepts can be used for audio indexing. Speaker diarization systems are designed to answer the question “who spoke when?” by finding segments in an audio stream that exhibit similar properties in feature space, i.e. sound similar. Using a diarization system, all the content of an audio file is analyzed and similar sounds are clustered. This article provides an in-depth analysis on the statistic properties of similar acoustic segments identified by the diarization system in a predefined document set and the theoretical fitness of this approach to discern one document class from another. It also discusses how diarization can be tuned in order to better reflect the acoustic properties of general sounds as opposed to speech and introduces a proof-of-concept system for multimedia event classification working with diarization-based indexing.

KEYWORDS: Computer Science, Multimedia, Audio Indexing, Speaker Diarization, Information Retrieval

1. Introduction

To tackle the challenge of indexing multimedia data, a multitude of approaches have been proposed in the last decade ((Lew et al. 2006) or (Snoek et al. 2009) for an overview). Most video contents contain both audio and visual information. Many approaches, however, focus only on the visual part of a video. Recently, audio has begun to play its part in multi modal multimedia analysis and can be leveraged to complement results from visual analysis techniques to increase the effectiveness of multimedia detection or retrieval approaches. Audio information can be engaged to these ends in two essentially different ways. Since the late 1990s (Wactlar et al. 1996), speech recognition has been used for video analysis. Detecting sound concepts, which describe a video’s content, is another way of using audio information for video content detection. Manually defined low-level acoustic concepts such as ″people laughing″ or ″indoor sound″ transport useful information to a video’s content and such concepts can be automatically recognized once a system is trained to detect them (Li et al. 2010) (Jiang et al. 2010). This approach does, however, usually involve manual concept definition. The two drawbacks of manual concept definition are that it introduces a human bias since human annotators are likely to identify these concepts based on different properties of sound than a machine algorithm would, and it usually leads to rather abstract

concepts. This paper explores the applicability of a speaker diarization engine to extract low-level acoustic concept from domain specific training data. A speaker diarization engine clusters segments of an audio stream that have similar properties. It automatically extracts acoustic concepts defined by a machine algorithm, namely the speaker diarization engine. To examine whether this assumption holds, we have generated and examined diarization data from the TRECVID 2011 development data set (NIST 2011), which has been released by NIST as training data for the TRECVID Multimedia Event Detection (MED) 2011 challenge. It contains randomly selected videos that are examples from fifteen different categories of high-level concepts such as "woodworking project" and "wedding ceremony", and thus represents a suitable data set for the analysis presented in this paper. The data set not only deliver a wealthy low-level features that can be detected by the diarization approach, but also separate data into higher-level classes such that we can explore whether higher-level classes can be predicted by the presence or absence of certain low-level features. The analysis of the distribution of speaker segments discovered in the videos from different categories implies that speaker segments are not randomly distributed but useful in predicting whether a video belongs to a certain class of event or not. It shows that speaker diarization generates low-level audio concepts that are helpful in higher-level event classification and detection. An in-depth analysis of the application of the speaker diarization engine to a group of sounds, selected from freesound.org and combined in a single sound, examines the performance of speaker diarization on a number of examples from five selected sound categories. This analysis was also used to tune parameters for speaker diarization. Finally, this paper presents an event detection system that utilizes speaker diarization for indexing audio contents. Finally, we present an event detection system that uses speaker diarization for indexing audio contents. For comparison purposes, we have evaluated the system performance on data from the NIST TRECVID 2010 MED corpus.

The remainder of this paper is organized as follows: Section 3 briefly explains the ICSI speaker diarization system. Section 4 presents an experiment applying diarization to the NIST TRECVID MED 2011 data set and a discussion of the experiment's findings. Section 5 presents an in-depth analysis of the performance of speaker diarization on a synthetically compiled sound set and discusses the implications for diarization tuning. Section 6 introduces a diarization-based event detection system and presents an evaluation of this system's performance on the NIST TRECVID 2010 MED corpus. The paper concludes with a discussion of the relevance of the findings of this paper as well as perspectives for further research.

2. Related Work

Many systems have applied audio based methods to multimedia indexing and analysis we will discuss some of the most relevant to our work. In video scene segmentation the use of audio features dates as early as 1999 (Huang 1999). Many approaches employ supervised learning techniques in which classifiers are trained to discover distinct low-level sound concepts such as "indoor sound" or "people laughing". The number of classifiers trained in these approaches ranges from ten as

described in (Jiang 2010) to 75 in an approach for video scene segmentation described in (Sidiropoulos 2011). The downside of these supervised approaches is that training data has to be manually selected for each new low-level sound-concept in order to train models for new application domains and that each low-level sound category has to be anticipated by the specialists that train the system. Unsupervised approaches like the ones presented in (Lu 2008) and (Chaudhuri 2011) generate models of the audio content analyzed by learning the structure features found in training material. Hence, in unsupervised approaches a specialist only needs to provide a set of training files for any new higher-level category to be learned by the system. A disadvantage of the majority of both supervised and unsupervised approaches is that they are only using a fraction of the audio material to account for automatic analysis. Only very few approaches take a more holistic perspective and use the entire audio contents of each file (Mertens 2011) and (Chaudhuri 2011). While holistic approaches sometimes suffer from an increased complexity in terms of time and space complexity, the approach presented in this paper reduces complexity at an early stage by computing abstract a video representation based on abstract sound concepts that are generated by clustering sound models from individual input videos.

The approach presented in this paper is a speaker diarization system, which automatically segments the whole content of each input file and then uses all of the segments found in all files indexed by the system. Another advantage of using speaker diarization is that the segments are presented by the diarization system in a way that is accessible by users allowing them to listen to the sounds identified by the system. To a certain extent, this feature brings diarization-based audio indexing and event detection closer to a supervised system: In supervised systems, the low-level sound concepts are defined by human users, in a diarization-based system they can at least be listened to. In future versions of the prototype system presented in this paper, we plan to give users the possibility to exclude low-level sound categories found by the system. This feature could be useful in the case of artifacts introduced by recording equipment or other misleading audio information.

The proof-of-concept system introduced in this paper to practically show the applicability of speaker diarization to audio indexing and audio-based event detection is most similar to the approaches described in (Lu 2008) and in (Chaudhuri 2011) . The system from Lu and Hanjalic (Lu 2008) is inspired by term frequency and inverse document frequency from text analysis. In contrast to the system introduced in this paper, the system from Lu and Hanjalic only extracts key segments for audio classification. The approach from (Chaudhuri 2011) employs audio segmentation and models segments as Hidden Markov Models. The approach presented in this paper differs from that approach in that it introduces a higher-level abstraction layer that provides an easier-to-analyze representation.

3. ICSI Speaker Diarization System

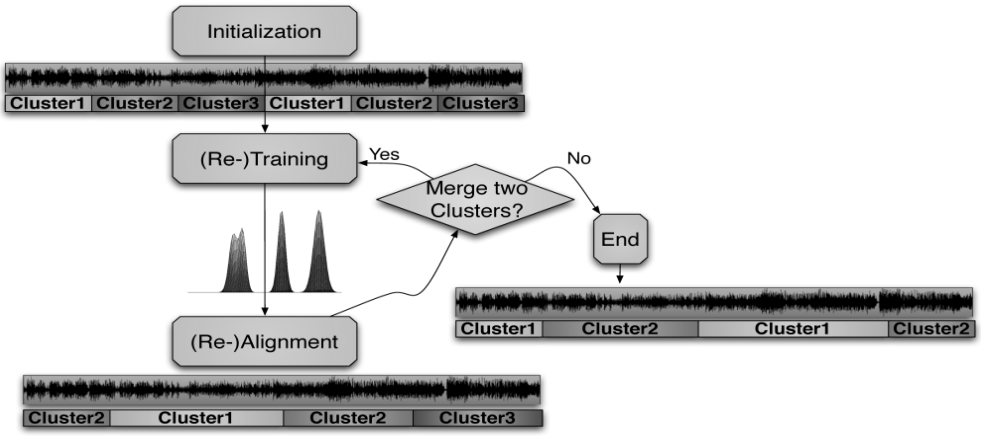

Figure 1: ICSI Diarization Engine

For detecting sound concepts in each individual video we used a system based on the ICSI speaker diarization system (Wooters 2008) in a faster-than-realtime version (Huang 2007). The actual diarization process consists of a pre-processing phase and a segmentation and clustering phase, as shown in figure 1.

In the preprocessing phase audio features, in this case Mel-Frequency Cepstral Coefficients (MFCCs) are extracted from the video soundtrack. We use a frame period of 10 ms with an analysis window of 30 ms in the feature extraction, a setting that we have previously used successfully in other audio based detection tasks. In its original application context the speaker diarization system also employs speech/non-speech segmentation to exclude non-speech in later processing steps. In order to use all audio information, we have omitted the exclusion of non-speech segments.

In the segmentation and clustering stage of speaker diarization, an initial segmentation is generated by uniformly partitioning the audio track into K segments of the same length. K is chosen to be much larger than the assumed number of speakers in the audio track. For meeting recordings of about 30 minute length, previous work (Imseng 2009) experimentally determined K = 16 a good value. For the audio tracks used in the TRECVID MED 2011 data set, we have determined K=64 a suitable value. The main reason for the higher K value is that the number of significant audio concepts in a video is much higher than the average number of speakers in a meeting video. The procedure for diarization is shown in Figure 1 and takes the following steps, a more detailed description can be found in (Wooters 2008):

1) **Initialization:** Train a set of Gaussian Mixture Models (GMMs), one for each initial cluster.

2) **Re-segmentation:** Re-segment the audio track using the current GMMs using majority vote on the likelihoods of a specified minimum duration (Friedland 2008). For the audio concept detection described in the next section, we have set this minimum duration to 200 milliseconds in order to capture sounds of smaller duration. This parameter selection was based on theoretical considerations. The experiments discussed in section 5 show that lower values would have been desirable. For speaker segmentation higher values are used. A typical minimum for speaker segmentation would be 2500 milliseconds

3) Re-training: Retrain the GMMs on current segmentation using the expectation-maximization (EM) algorithm (Friedland 2008).

4) Agglomeration: Select the closest pair of clusters and merge them. At each iteration, the algorithm checks all possible pairs of clusters to see if there is an improvement in BIC (Bayesian Information Criterion) scores by merging each pair and re-training it on the combined audio segments. The clusters from the pair with the largest improvement in Bayesian Information Criterion (BIC) scores are merged and the new GMM is used. The algorithm then repeats from the re-segmentation step until there are no remaining pairs that will lead to an improved BIC score.

The result of this algorithm consists of a segmentation of the audio track with n clusters and with one GMM for each cluster, where n is assumed to be the number of speakers, which in our case are audio concepts.

4. Data Analysis on TrecVid MED 2011 Data Set

4.1 TRECVID MED 2011 Development Data Set

The experiments detailed in this work were primarily done using the TRECVid 2011 Med dataset. This dataset is comprised of consumer-produced videos collected from social networking sites, or "found videos." The videos in this set are relatively short, usually no more than a couple of minutes and are produced by amateurs. The data is broken down into 15 categories, "event-kits", with 5 of those categories available in the test set. The training set is comprised of 2040 videos, and the test set of 4251. The event categories available in the test set are "attempting a board trick'", "feeding an animal'", "landing a fish", "wedding ceremony", and "working on a woodworking project". Of the test set 496 videos are from these 5 categories, and the remainder 3755 videos are random videos not belonging to any of the event categories. The exact breakdown of videos by category is presented in Table 1. Within a given category there are a wide variety of videos; the "wedding ceremony" event kit contains videos of a catholic mass, a Hindi ceremony, home made music videos, and an outdoor wedding, done in a snowy area, with a dogsled team passing in the background. Similarly the "attempting a board trick" event includes people skateboarding, surfing, and snowboarding.

The results and analysis presented here are based on the training set of the TRECVid 2011 Med dataset. Human analysis of a subset of this data has shown a wide variety of different types of sound, many of which are acoustically similar to the sounds perceived in the official categories (for example, power tools have a similar sound, regardless of their use in woodworking, metalworking, or in a factory). In addition, it is not uncommon for the audio track to be replaced completely from what was originally present, for example a home-made music video.

Category	Description	Train Data	Test Data

E001	Board Tricks	160	111
E002	Feeding Animal	160	111
E003	Landing Fish	122	86
E004	Wedding	128	88
E005	Woodworking	142	100
E006	Birthday Party	173	0
E007	Changing Tire	110	0
E008	Flash Mob	173	0
E009	Vehicle Unstuck	131	0
E010	Grooming animal	136	0
E011	Make a Sandwich	111	0
E012	Parade	134	0
E013	Parkour	108	0
E014	Repair Appliance	123	0
E015	Sewing	116	0
Rest	Random Other	N/A	3755

Table 1: Number of Videos for Train and Test

4.2 METHODOLOGY

We used the ICSI speaker diarization system with the TRECVID MED 2011 data set, to discover whether speaker diarization would aid in audio concept detection. The concept of this approach is that the speaker segments or "clusters" created with our diarization tools would exhibit similar acoustic properties to a speaker model. It was necessary for us to remove the standard preprocessing filters such as speech-nonspeech to produce models that would represent a low level audio concept, rather than a individual speaker. The hope was that certain types of sounds, such as a car sounds in a tire change video, or a power drill in action in a video about woodworking would be discovered.

We use Gaussian Mixture models to represent the audio concepts (which in a standard system would be speaker models) in our system. A Gaussian Mixture Model, or GMM, is a set of Gaussian distributions that describe each feature in the model, as shown in equation 1.

$$p(\vec{x}|\lambda) = \sum_{i=1}^{M} w_i N(\vec{x}|\mu_i, \Sigma_i) \quad (1)$$

In the equation $\vec{x}$ is a D-dimensional random vector, $N(\vec{x}|\mu_i, \Sigma_i)$ $i = 1,\ldots, M$, are the component densities and w_i, $i=1,\ldots, M$, are the mixture weights. Each component density is a D-variate Gaussian function of the form with mean vector μ_i and covariance matrix Σ_i (we use diagonal covariance matrix here). The mixture weights are constrained by $\sum_{i=1}^{M} w_i = 1$.

A single feature is represented by a number of Gaussians that are weighted according as to how they influence the overall model. The Gaussian distributions are represented by their variance and their mean value.

In order to match low level audio concepts across training videos and to also be able to classify low level feature models found in testing videos, we have simplified the Gaussian mixture model per speaker to a single vector that consists of the sums of the weighted means and the sums of the weighted variances of each Gaussian. In the remainder of this paper, we will call this vector a simplified supervector, as shown in equation 2.

$$W(x) = [\sum_{i=1}^{M} w_i m_i, \sum_{i=1}^{M} w_i S_i] \quad (2).$$

A Kmeans method was then used to cluster the simplified supervectors that were generated from all of the low level acoustic concepts, resulting in clusters that represent abstractions of the simplified supervectors of those concepts. These can then be mapped back to the concepts in each video by calculating the distance between the video's speaker models and the abstract simplified supervectors. This remapping allows us to count the overall occurrences of an individual abstract acoustic low level concept in all videos. We did this calculation, and calculated the number of occurrences of each abstract low level feature in each of the event sets. This allows us to compute the normalized frequency of each low level acoustic concept per event as shown in equation 3.

$$EEH(c_i, E) = \frac{\Sigma_k \Sigma_j n_j P(c_i = c_j \mid c_j \in D_k \cap D_k \in E)}{\Sigma_k \Sigma_j n_j P(c_i = c_j \mid c_j \in D_k)} \quad (3)$$

where EEH represents the expected event histogram and n_j is the occurrence number of c_j in audio clip D_k. $P(c_i=c_j|c_j \in D_k, D_k \in E)$ is the probability of audio term c_i equal c_j given c_j is in the audio clip D_k in the event E, and $P(c_i=c_j|c_j \in D_k)$ is the probability of audio term c_i equal c_j given c_j is from audio clip D_k.

The higher the normalized frequency of a sound belonging to an abstract acoustic low level concept a in an event b is, the higher is the predictive power of these concepts. For example, if we could create a model or models that would identify that would identify the sound of a power drill with perfect accuracy, and we assert that this sound would only occur in videos of the category "woodworking," we would have a normalized frequency of 100%. In essence, whenever we find the sound of a power drill, we know we have a woodworking video.

We have extracted the low level acoustic concepts with the most predictive power per event, and subjected them to a much closer examination, to determine if speaker diarization tools will help us here. We discuss this more thoroughly in the results section.

In figure 2 we detail the workflow of the system, consisting of two parts: the ICSI speaker diarization system and Kmeans clustering. As an example suppose we have three audio clips: 1, 2, and 3 (as shown in the top portion of the figure).

The ICSI speaker diarization system segments each clip into separate chunks. If these chunks exhibit similar acoustic properties according to the system, they will be considered to belong to the same speaker. For example, in audio clip 1, the clustering created speakers A, B, and C. Note that

each speaker in each audio clip is described by a Gaussian Mixture Model.

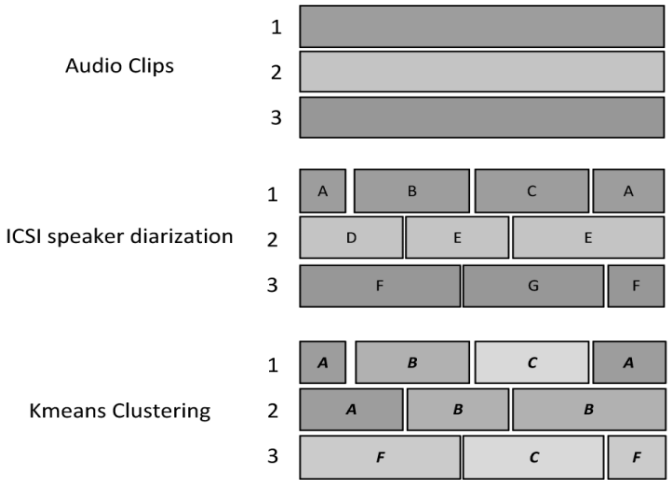

Figure 2: Workflow for speaker diarization and clustering.

Figure 2 gives us information about the output of the ICSI speaker diarization system. We use the simplified supervector we obtained from variance and weighted mean to represent a speaker. Then we use Kmeans clustering to cluster speakers from all audio clips. In figure 2 we see that speaker A in clip 1 and speaker D in clip 2 are clustered together.

4.3. Results from Data analysis

The acoustic low level concepts produced by our system were then analyzed with regards to their predictive power for the abstract concepts presented in the event kits (board tricks, changing a tire, woodworking, etc.) that were found in the training sets. Given that there are 15 categories possible, chance would be 6.7%, however we found 5 abstract concepts which had a normalized frequency of 50% or greater: 100% for E001, 71.1% for E005, 71.4 % for E007, 64.28 % for E011, 50 % for E004. In other words, we can correctly determine the category for one of these events with a high degree of accuracy simply by the presence of an instance of one of these low level concepts. Not all events lent themselves to such high accuracy, with Changing a tire performing worst, with only a 34% accuracy based upon its dominant sound concept. Overall, it seems that the sound concepts we generate with the speaker diarization system are a good discriminator for the higher level concepts found in the 15 event classes in the NIST TRECVID 2011 data set. In table 2 we see the normalized frequencies of the top five acoustic low level concepts for all events in the training kit.

We generated the normalized frequencies with kMeans clustering, with k equaling 200. This produced and average normalized frequency for the best low level sound concept of 46.6% accroos all events. When we used k=100 for our clustering this droped to 39.8%, and if we used k=1000 we only achieved a 1% average normalized frequency, probably due to over fitting to specific sound concepts from individual videos.

Category					
E001	100% (ID 153)	25.8% (ID 117)	25% (ID 130)	19% (ID 189)	18.9% (ID 10)
E002	16.9% (ID 85)	14.8% (ID 108)	14.7% (ID 129)	14.6% (ID 90)	14.2% (ID 184)
E003	40.00% (ID 188)	31.25% (ID 13)	25.58% (ID 18)	22.32% (ID 58)	16.26% (ID 133)
E004	50.00% (ID 47)	41.66% (ID 48)	35.86% (ID 175)	33.33% (ID 94)	30.95% (ID 149)
E005	71.7% (ID 161)	58.3% (ID 88)	25% (ID 130)	24.7% (ID 66)	19.9% (ID 54)
E006	40.0% (ID 188)	14.2% (ID 52)	13.6% (ID 103)	13.3% (ID 4)	13% (ID 71)
E007	71.4% (ID 62)	43.5% (ID 51)	42.8% (ID 199)	41.9% (ID 139)	41.6% (ID 186)
E008	37.4% (ID 178)	29.8% (ID 110)	28.1% (ID 66)	25.0% (ID 130)	21.9% (ID 79)
E009	37.41% (ID 178)	29.81% (ID 110)	28.08% (ID 66)	25.00% (ID 130)	21.91% (ID 79)
E010	25.00% (ID 13)	21.42% (ID 169)	15.00% (ID 157)	14.91% (ID 65)	14.77% (ID 82)
E011	64.28% (ID 169)	36.36% (ID 103)	33.33% (ID 94)	32.98% (ID 70)	23.94% (ID 40)
E012	40.00% (ID 22)	40.00% (ID 156)	34.33% (ID 186)	33.33% (ID 135)	30.00% (ID 176)
E013	23.68% (ID 76)	15.11% (ID 43)	14.24% (ID 2)	13.07% (ID 20)	12.67% (ID 17)
E014	31.42% (ID 177)	30.43% (ID 84)	27.84% (ID 106)	27.08% (ID 98)	24.61% (ID 152)
E015	41.66% (ID 48)	33.33% (ID 94)	19.71% (ID 40)	14.77% (ID 37)	14.28% (ID 62)

Table 2: Top five sound concepts according to normalized frequency

It is quite encouraging to find sounds that have very high predictive power; however we must even that with knowledge that they may be highly infrequent. For example Sound 153, which only occurs once in the entire training set. It was described by the annotator as a "fast gurgling water sound," and it occurs in the board trick category in a video about surfing. The top sound concept in E005, number 159, occurs in 21 videos, with only 12 of them actually in E005. Manual inspection of sound 159 by our annotator found that the sound was comprised primarily of engine sounds of moderate volume. The sound occurs in event categories E001, E009, E014, and E015 aside from E005. To show the wide variety of sounds that one sound concept can represent, we note that in one occurrence in E001 we have a harp sound, and in another we have the sound of an air compressor. In E009 it occurs in only two videos, as the sound of racing car engines a moderate distance away. In E015 this sound is in 4 videos, 3 of which are the sound of a sewing machine, which is acoustically quite similar to the engine sounds of E005, with the other being the sound of some kind of rotary engine.

We also observed several sound concepts for each specific event class that are not present in any video of that class. These numbers are E001: 35, E002: 29, E003: 27, E004: 7, E005:20, E006: 11, E007: 32, E008: 11, E009: 28, E010: 14, E011: 15, E012: 27, E013: 23, E014: 66 and E015: 22.

These observations suggest that the presence or absence and number of times a sound concept occurs in a given video are predictors for the event category of said video.

In Figure 3 we see the distribution of sound concepts in the test set according to their frequency of

occurrence for clustering with the kMeans clustering values of k=100, k=200, k=300 and k=1000. The figures suggest that as we increase the k value in the clustering, the distribution more closely resembles a Zipfian distribution. Zipfian distributions are sometimes connected to the applicability of TF-IDF measures, even though this is controversial from a theoretical perspective (Robertson 2004).

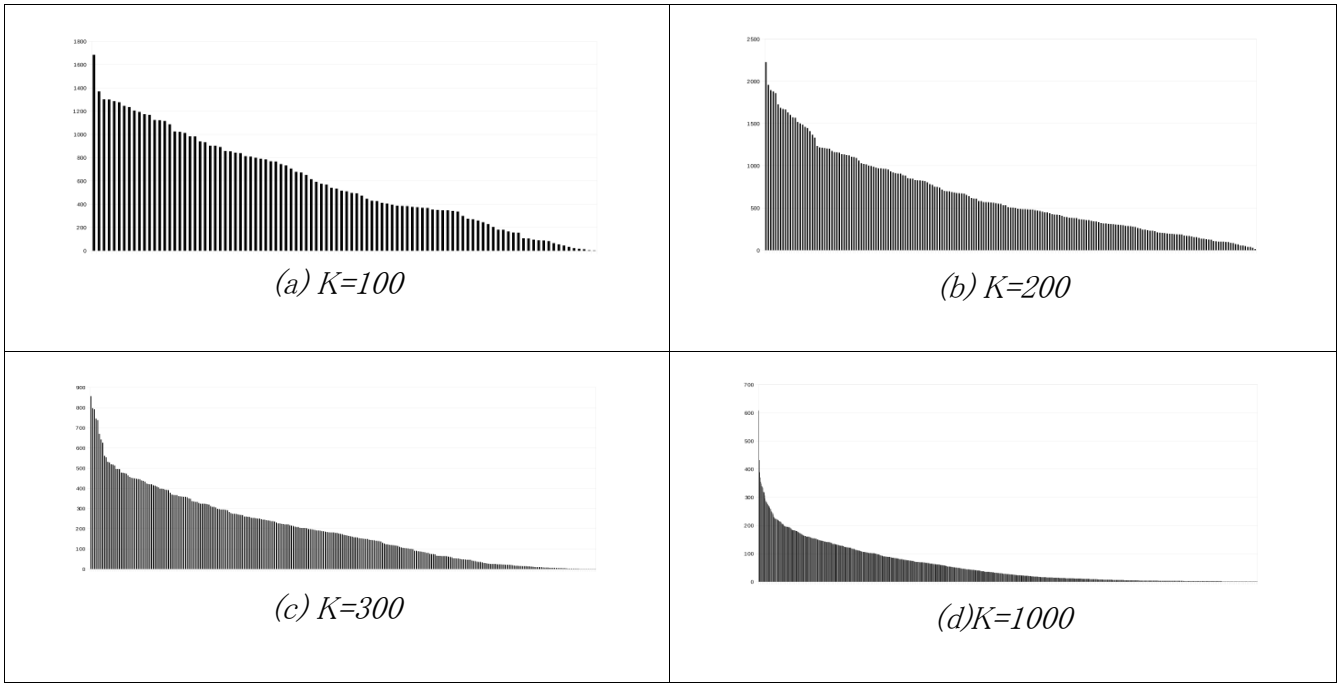

Figure 3: Distribution Frequency for Kmeans K=100, 200, 300, 1000.

Beyond the analysis of the sound concepts that performed best according to our measures, we also annotated and studied the distributions of seven other sound sets, without selecting for performance. Sound 5 turned out to have poor discriminative powers, but was still quite useful: it occurs in every sound category, although primarily in E004, E005, and E008, together representing 41% of its occurrences. Our annotator discovered that it was most often the sound of guitar music and singing, and for this type of sound it was a good discriminator. This cluster also contained speech, machine noises, and even the sound of bacon cooking. Sound 8 seems to represent music with percussive elements in E0014, although in a in a single event with a 10% or greater chance of occurrence. In E014 this turned out to be tool use with music in the background, but in other events it was a range of sounds from cars moving at speed, to bass heavy techno. A final interesting example is sound 76, which to a human annotator sounded to be mostly instrumental music, which heuristically could be expected to occur quite often, as music is a common component across all categories; however it only appeared in 25 videos, and 9 of 15 event classes.

5. Diarization Tuning

The general statistical observations presented in the previous chapter have indicated that sound diarization can be used to generate discriminative representations of videos on the level of sound

distributions. In order to improve the discriminative power of these representations, sound diarization can be tuned in a number of ways. The most obvious parameters in this endeavor are the minimum segment length of audio segments found by diarization and the number of disparate sound classes to be searched for per video. The experiment described in this section studies the effects of tuning these parameters by an in depth analysis of the performance of different parameter settings on a single audio file with example sounds for the five sound categories.
In the experiment discussed in the previous section we attempted to classify data from TRECVID, MED 11 which has sets of videos classified into various categories, such as woodworking, landing a fish, or performing a board trick. A human annotator listening to the audio tracks of these videos would have difficutly classifying them in many cases. This motivated us to develop a small data set that would be trivial for a human to classify. We compiled this data with samples acquired from FreeSound.org's open data set. The categories which we used were dogs barking, cars, church bells, guns, rivers, and trains. These were selected because our human annotator found them to be easy to distinguish, and because FreeSound had a sufficient number of high quality samples in each category.

After determining which categories to use, and acquiring six samples of each category, we randomized the order of the tracks and concatenated them into a single twenty-four minute long track, which we attempted to segment with our diarization system. We randomly added three extra tracks from our set to this audio file, to determine if when clustering occurred if our system would put those into the same or different clusters. The sound instances were compiled into one single audio file. While these sound instances were semantically labeled, they do not necessarily contain the same sounds. We arranged the sounds in the sound file in the order presented in table 3.

File name	Category	Start	Duration
11686_bram_05-bells-on-market-in-steenokkerzeel	church-bells	0	58.23
17008_laurent_spring-river-2	rivers	58.23	26.86
1926_rhumphries_rbh-le-mans-passby-01	cars	85.09	3.46
17366_cognito-perceptu_train-passing-cars	trains	88.55	38.4
17851_patchen_locomotive-1-distant-horn	trains	126.95	23.66
107621_stintx_gunshot-04	guns	150.62	2.1
17741_krisboruff_babbling-brook	rivers	152.71	15.45
1934_rhumphries_rbh-train-freight-by	trains	168.16	196.05
110622_soundscalpel-com_warfare-gunshots-machine-gun-burst-001	guns	364.31	2.86
11686_bram_05-bells-on-market-in-steenokkerzeel	church-bells	367.07	58.23
24965_mich3d_bigdogbarking-02	barks	425.31	3.75
25068_laurent_locomotive	trains	429.06	144.75
3179_pingel_passingcar01	cars	573.81	13.57
13571_acclivity_dommebells	church-bells	587.38	68.61
30344_ronfont_my-dog-george	barks	655.99	18.47

19493_inchadney_mountainstream	rivers	674.46	94.55
20204_inchadney_harzmountain-stream	rivers	769.01	121.51
25039_dobroide_20061105-gunshot-03	guns	890.53	3.95
50910_rutgermuller_in-car-driving	cars	894.48	52.34
40154_genghis-attenborough_church-bells	church-bells	946.82	129.63
34872_sruddi1_bark1	barks	1076.45	9.61
4912_noisecollector_barks	barks	1086.06	2.69
67280_erh_collob-church-bells-2	church-bells	1088.75	42.87
7876_sazman_train-passing-2	trains	1131.62	38.83
30749_matt-g_m240	guns	1170.45	12.76
28170_herbertboland_beek3	rivers	1183.21	32.64
51054_ingsey101_driving-sounds	cars	1215.84	4.52
71741_audible-edge_nissan-maxima-handbrake-turn-04-25-2009	cars	1220.36	10.37
9032_mistertood_dog-bark2	barks	1290.73	2.83
30749_matt-g_m240	guns	1233.62	1246.37
40154_genghis-attenborough_church-bells	church-bells	1259.13	129.63

Table 3: Sounds in synthetically compiled sound file

In the experiment we analyzed the segmentation performance delivered by several parameter settings and then manually inspected the sound classes found by the diarization algorithm. Table 4 shows the distribution of noise segmentation on the test data using six classes and a minimal segment length of 256 milliseconds in a confusion matrix like representation. The numbers in each cell of the confusion matrix give the percentage of the respective clip assigned to a certain class. For instance the second line of Table 4(a) means that five percent of clip 4912 were identified as belonging to class 2 while 95 percent of that clip were identified as belonging to class 0. This data shows that six classes are not enough to identify the six sound concepts given in the test data as some sound concepts like guns or rivers are sometimes classified as belonging to class 5 and 0 or 3 and 0 respectively. We have hence explored a differing number of classes ranging from 2 to 512. We have found that for the test data 16 classes seem to result in a useful size that allows for differentiating between the target sound concepts from the test data.

We have also explored disparate minimal segment length values. Table 4(b) shows the results for 16 classes with a minimal segment length of 256 milliseconds. In class 14, many sound instances from different sound categories can be found, many of them fitting into this class at 100%. A closer look at the properties of the sound instances, especially the barks and gunshots reveals that all sounds have similar parts that are significant to the clustering algorithm. The minimum segment length then forces neighboring frames to be clustered in the same sound class. This effect can be mitigated by

sm3_6_256.mfcc.rttm	0	1	2	3	4	5
barks 34872_sruddi1_bar...	0	0	100	0	0	0
barks 4912_noisecollector_...	95	0	5	0	0	0
barks 24965_mich3d_big...	0	0	66	0	0	34
barks 9032_mistertood_d...	0	0	94	0	0	6
barks 30344_ronfont_my-...	0	0	100	0	0	0
cars 3179_pingel_passingc...	36	0	1	63	0	44
cars 1926_rhumphries_rb...	0	0	0	0	0	100
cars 50910_rutgermuller_i...	6	94	0	0	0	0
cars 71741_audible-edge_...	0	0	25	0	0	75
cars 51054_ingsey101_dri...	7	0	73	0	0	20
church-bells 40154_genghi...	0	21	0	0	79	0
church-bells 67280_erh_c...	0	0	0	0	100	0
church-bells 13571_acclivit...	0	0	98	0	0	2
church-bells 40154_genghi...	0	21	0	0	79	0
church-bells 11686_bram_...	19	73	0	0	0	8
church-bells 11686_bram_...	19	73	0	0	0	8
guns 30749_matt-g_m240...	0	0	0	0	0	100
guns 110622_soundscalpel...	0	13	0	0	0	87
guns 107621_stintx_gunsh...	0	0	0	0	0	100
guns 25039_dobroide_20...	100	0	0	0	0	0
guns 30749_matt-1g_m2...	0	0	0	0	0	100
rivers 17008_laurent_sprin...	68	0	0	0	0	32
rivers 19493_inchadney_m...	0	0	0	100	0	0
rivers 28170_herbertbolan...	100	0	0	0	0	0
rivers 20204_inchadney_2...	0	0	0	100	0	0
rivers 17741_krisboruff_b...	99	0	0	0	0	1
trains 1934_rhumphries_1_r...	84	13	0	0	0	3
trains 17851_patchen_loco...	32	0	11	0	0	57
trains 7876_sazman_train...	39	0	0	0	0	61
trains 25068_laurent_1loc...	29	30	7	13	0	21
trains 17366_cognito-perce...	27	0	0	73	0	0

Table 4(a): 6 classes with 256 ms minimal length

sm3_qb_16_256.mfcc.rttm	0	1	2	3	5	6	7	9	10	11	13
barks 34872_sruddi1_bar...	0	0	0	0	0	0	100	0	0	0	0
barks 4912_noisecollector_...	95	0	0	0	0	0	5	0	0	0	0
barks 24965_mich3d_big...	0	0	0	0	0	0	100	0	0	0	0
barks 9032_mistertood_d...	0	0	0	0	0	100	0	0	0	0	0
barks 30344_ronfont_my-...	0	0	0	0	0	0	100	0	0	0	0
cars 3179_pingel_passingc...	0	0	0	0	0	100	0	0	0	0	0
cars 1926_rhumphries_rb...	0	0	0	0	0	100	0	0	0	0	0
cars 50910_rutgermuller_i...	0	0	5	0	0	0	0	0	95	0	0
cars 71741_audible-edge_...	0	0	0	0	28	72	0	0	0	0	0
cars 51054_ingsey101_dri...	0	0	0	0	0	100	0	0	0	0	0
church-bells 67280_erh_c...	0	0	0	0	0	0	67	0	0	0	33
church-bells 13571_acclivit...	0	0	0	0	0	2	98	0	0	0	0
church-bells 40154_genghi...	0	0	0	0	0	0	3	0	0	97	0
church-bells 11686_bram_...	100	0	0	0	0	0	0	0	0	0	0
guns 30749_matt-g_m240...	0	0	0	0	0	100	0	0	0	0	0
guns 110622_soundscalpel...	0	0	0	0	0	100	0	0	0	0	0
guns 107621_stintx_gunsh...	0	0	0	0	0	100	0	0	0	0	0
guns 25039_dobroide_20...	0	0	0	0	0	66	0	0	34	0	0
rivers 17008_laurent_sprin...	88	0	0	0	0	12	0	0	0	0	0
rivers 19493_inchadney_m...	0	100	0	0	0	0	0	0	0	0	0
rivers 28170_herbertbolan...	0	0	0	0	0	1	0	0	0	0	99
rivers 20204_inchadney_h...	0	0	0	0	0	0	0	90	10	0	0
rivers 17741_krisboruff_b...	0	98	0	0	0	2	0	0	0	0	0
trains 17851_patchen_loco...	0	99	0	0	0	1	0	0	0	0	0
trains 25068_laurent_loco...	0	0	0	0	55	45	0	0	0	0	0
trains 1934_rhumphries_r...	16	0	43	41	0	0	0	0	0	0	0
trains 7876_sazman_train...	6	0	0	0	0	0	0	0	0	0	93
trains 17366_cognito-perce...	0	100	0	0	0	0	0	0	0	0	0

Table 4(b): 16 classes with 256 ms minimal length

sm3_qb_16_128.mfcc.rttm	0	1	2	3	5	6	7	8	9	11	13
barks 34872_sruddi1_bar...	0	0	0	0	0	100	0	0	0	0	0
barks 4912_noisecollector_...	0	0	0	0	0	100	0	0	0	0	0
barks 24965_mich3d_big...	0	0	0	0	0	100	0	0	0	0	0
barks 9032_mistertood_d...	0	0	0	0	0	100	0	0	0	0	0
barks 30344_ronfont_my-...	0	0	0	0	0	9	91	0	0	0	0
cars 3179_pingel_passingc...	0	0	0	0	0	100	0	0	0	0	0
cars 1926_rhumphries_rb...	0	0	0	0	0	63	0	0	0	0	37
cars 50910_rutgermuller_i...	0	94	4	0	0	1	0	0	0	0	0
cars 71741_audible-edge_...	0	0	0	0	77	23	0	0	0	0	0
cars 51054_ingsey101_dri...	0	0	0	0	0	100	0	0	0	0	0
church-bells 67280_erh_c...	0	0	0	0	0	67	0	0	0	0	33
church-bells 13571_acclivit...	0	0	0	0	0	3	97	0	0	0	0
church-bells 40154_genghi...	0	0	0	0	0	5	0	0	0	95	0
church-bells 11686_bram_...	100	0	0	0	0	0	0	0	0	0	0
guns 30749_matt-g_m240...	0	0	0	0	0	100	0	0	0	0	0
guns 110622_soundscalpel...	0	0	0	0	0	100	0	0	0	0	0
guns 107621_stintx_gunsh...	0	0	0	0	0	100	0	0	0	0	0
guns 25039_dobroide_20...	0	33	0	0	0	67	0	0	0	0	0
rivers 17008_laurent_sprin...	93	4	0	0	0	2	0	0	0	0	0
rivers 19493_inchadney_m...	0	0	0	0	0	0	0	100	0	0	0
rivers 28170_herbertbolan...	0	0	0	0	0	1	0	0	0	0	99
rivers 20204_inchadney_h...	0	11	0	0	0	0	0	0	89	0	0
rivers 17741_krisboruff_b...	0	99	0	0	0	1	0	0	0	0	0
trains 17851_patchen_loco...	0	79	0	0	20	1	0	0	0	0	0
trains 25068_laurent_loco...	0	1	0	1	60	36	0	0	0	0	1
trains 1934_rhumphries_r...	15	0	42	43	0	0	0	0	0	0	0
trains 7876_sazman_train...	0	3	0	0	0	0	0	0	0	0	97
trains 17366_cognito-perce...	0	100	0	0	0	0	0	0	0	0	0

Table 4(c): 16 classes with 128 ms minimal length

sm3_16_32.mfcc.rttm	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
barks 34872_sruddi1_bar...	0	0	0	0	0	0	0	0	0	0	0	0	0	39	56	5
barks 4912_noisecollector_...	0	0	0	0	0	0	41	0	0	0	0	0	0	0	44	16
barks 24965_mich3d_big...	0	0	0	0	0	0	2	0	0	0	0	0	0	45	53	0
barks 9032_mistertood_d...	0	24	0	0	0	0	0	0	0	0	0	0	0	0	76	0
barks 30344_ronfont_my-...	0	0	0	0	0	0	5	0	0	0	0	0	38	0	55	2
cars 3179_pingel_passingc...	0	96	0	0	0	0	1	0	0	0	0	0	0	0	1	2
cars 1926_rhumphries_rb...	0	0	11	48	0	0	0	0	0	0	0	0	0	0	13	29
cars 50910_rutgermuller_i...	0	0	2	1	0	0	0	0	0	0	96	0	0	0	1	0
cars 71741_audible-edge_...	0	4	0	0	0	0	73	0	0	0	0	0	0	0	20	4
cars 51054_ingsey101_dri...	0	16	0	0	0	0	0	0	0	0	0	0	0	0	84	0
church-bells 67280_erh_c...	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
church-bells 13571_acclivit...	0	0	0	0	0	0	2	97	0	0	0	0	0	0	1	0
church-bells 40154_genghi...	0	0	0	0	0	0	0	0	0	0	0	78	7	0	0	14
church-bells 11686_bram_...	54	0	0	0	39	0	6	0	0	0	0	1	0	0	0	0
guns 30749_matt-g_m240...	0	0	0	0	0	0	24	0	0	0	0	0	0	0	44	32
guns 110622_soundscalpel...	0	0	0	0	0	0	19	0	0	0	0	0	0	0	70	11
guns 107621_stintx_gunsh...	0	0	0	0	0	0	71	0	0	0	0	0	0	0	10	19
guns 25039_dobroide_20...	0	0	0	0	0	0	45	0	0	0	0	21	0	0	34	0
rivers 17008_laurent_sprin...	0	0	0	0	0	0	98	0	0	0	0	0	0	0	2	0
rivers 19493_inchadney_m...	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
rivers 28170_herbertbolan...	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
rivers 20204_inchadney_h...	0	0	0	0	0	0	0	0	0	91	8	0	0	0	0	0
rivers 17741_krisboruff_b...	0	2	8	0	0	0	10	0	0	0	0	0	0	0	80	0
trains 17851_patchen_loco...	0	33	0	0	0	0	35	0	0	0	0	0	0	9	1	23
trains 25068_laurent_loco...	0	15	5	1	0	23	44	0	0	0	7	0	0	0	4	1
trains 1934_rhumphries_r...	0	0	26	58	0	0	1	0	0	0	0	0	0	0	1	15
trains 7876_sazman_train...	0	15	4	1	0	0	3	0	0	0	0	0	0	68	1	8
trains 17366_cognito-perce...	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4d): 16 classes with 32 ms minimal length

Table 4: Results for mfcc-based noise segmentation

lowering the minimal segment length. As shown in Table 4(c), even a minimum segment length of 128 milliseconds does not suffice to discriminate between two different sound categories. While a minimum segment length of 64 milliseconds performs better, we found a minimum segment length of 32 milliseconds (Table 4(d)) gave more discriminating results.

We then ran our system with this combined sound file as its input, and had a human annotator describe the clusters, labeled as SPKR_0 to SPKR_15 in table 4(d) (All of the human annotation was done for the 16 cluster 32ms test case). Category 0 was the sound of a ringing bell's offset. Category 1 was a collection of sounds, non-variant soft machine whirs, wind, a horn, and water; all of the sounds were of mid energy, and were described as monotone. Category 2 was mostly the sound of a river, although a few other high energy low pitched sounds were include, such as a train rattling. Category 3 was a high energy machine sound. Category 4 was the complement to Category 0, the same bells onset, including the sound of the clapper hitting the bell. Category 5 was a fast paced bell from a train crossing, and as such was officially part of the Train set, rather than the church bell set. Category 6 was a low energy mid-pitch collection of sounds from many different categories, similar to a silence model. Category was a church bell. Category 8 was a different river

sound. Category 9 was birdsong. Category 10 was a collection of high pitched mid-energy sounds, such as car engines, fast birdsong, and train whistles. Category 11 was a carillon of church bells ringing, and a single gun shot. Category 12 was a specific dog bark, and church bells, almost entirely from a specific church. Category 13 was a mid energy train sound, a bark's offset and the wind. Category 14 was a widely variant collection of high energy sounds, including barks, gunshots, tire screeches, and a river. Category 15 was mostly silence, although a faint train whistle also found its way in.

In comparison to our work with the TrecVid MED11 videos, described above, in which we analyzed the best discriminating sound clusters, in this experiment we analyzed all of the sounds clusters, including those which were not shown to be discriminatory. Additionally this analysis is not based on a second level clustering as our MED11 analysis was, and we used a much smaller data set. With this in mind, it is interesting to note how the features we selected diarized the audio track. It looks as though our system is relatively good at discovering bells for example, however it doesn't create a single category (or speaker) for all of them, rather it tends to classify each bell as a separate category, or in the case of the audio file discussed in figure 4 the onset and offset are considered separate categories.

figure 4: 11686_bram_05-bells-on-market-in-steenokkerzeel

SPKR_0 and SPKR_4 only occur in this sound file. The other two sound categories occurring were SPKR_6 and SPKR_11 (albeit very briefly). SPKR_6 was a very broad model that included many low energy sounds, and SPKR_11 also contained church bells. It is worth noting that these categorizations occurreds towards the end of the audio clip when the bell ringing was tapering off, or over in the case of SPKR_6.

In figure 5 we have a river's sound:

figure 5: 17741_krisboruff_babbling-brook

At first glance it looks as the SPKR_14 is a good model of this river, however, our analysis has shown that it is a very broad cluster which describes sounds from many different categories. Many of river sounds did have highly distinct sound cluster categorization. If we were attempting to use this data for training purposes we would probably throw out SPKR_14, relying on the other single-

river sound categories to categorize an unknown river. The remainder of the sound categories were SPKR_6, our low energy catch-all model, SPKR_1 which was a low energy broad category, and SPKR_2, which was mostly river sounds.

6. Experiments on MED 10 Data Set

6.1. System Overview

In order to verify the applicability of diarization to audio indexing we have build a proof-of-concept system that uses a diarization based sound description for concept classification. An overview of the system is given in figure 6. The clustering step is shown in the lower left part of the figure. After sound feature extraction for each file 1,…,n, the system generates diarization models for each sound file (depicted as noise diarization 1a,…,1c for video 1 etc.). From these diarization models, supervectors are extracted and then clustered using Kmeans clustering. The resulting abstract sounds are then used to index each sound file by finding those abstract sounds that are closest to the supervector representation of the file's diarization models. For each sound file a description vector is generated. This vector holds one position for each abstract sound, so for k=n the vector has n elements.

We then train a Support Vector Machine (SVM) with intersection kernel for the multi-class event retrieval/classification problem using the event label information (E001 - E015). We used the LIBSVM implementation (Chang & Chih-Jen 2011). As input for the clustering itself we use all files available in the training and in the testing set. For the description vector, we have implemented a number of weighting schemes adapted from (Lu & Hanjalic 2008). In an analogy to text analysis, we can think of each clustered ID as an audio word and each audio clip as an audio document.

Term frequency (TF) is defined in equation (4).

$$TF(c_i, D_k) = \frac{\sum_j n_j P(c_i = c_j \mid c_j \in D_k)}{\sum_j} \quad (4)$$

where TF(c_i, D_k) is the term frequency of audio term c_i in the audio document D_k. $P(c_i = c_j | c_j \in D_k)$ is the probability that audio term c_i equals c_j in document D_k.

Similar to inverse document frequency (IDF) in text document analysis, IDF of an audio term can be defined as the log of the number of all documents divided by the number of documents containing the audio element.

$$IDF(c_i) = \log \frac{|D|}{\sum_k P(c_i \in D_k)} \quad (5)$$

where $|D|$ means the total number of documents and $P(c_i \in D_k)$ is the probability of term c_i in document D_k. Some text analysis applications have benefited from the use of logTF (Leopold 2002; Lan 2005); therefore these features are also considered in our experiments. Finally, for each audio document, we weight each audio element according to the indicators mentioned above, assuming indicators are independent of each other. For each audio clip D_k, we can represent it by the feature vector

$TF(Dk) = [\ TF(c_1;Dk),\ldots,\ TF(c_M;D_k)]$, (6)

and similarly for the feature vectors $\log TF(D_k)$, $TFIDF(D_k)$, and $\log TFIDF(D_k)$, etc, where the combination terms are the product of weights of individual indicators, and M is the total number of

words.

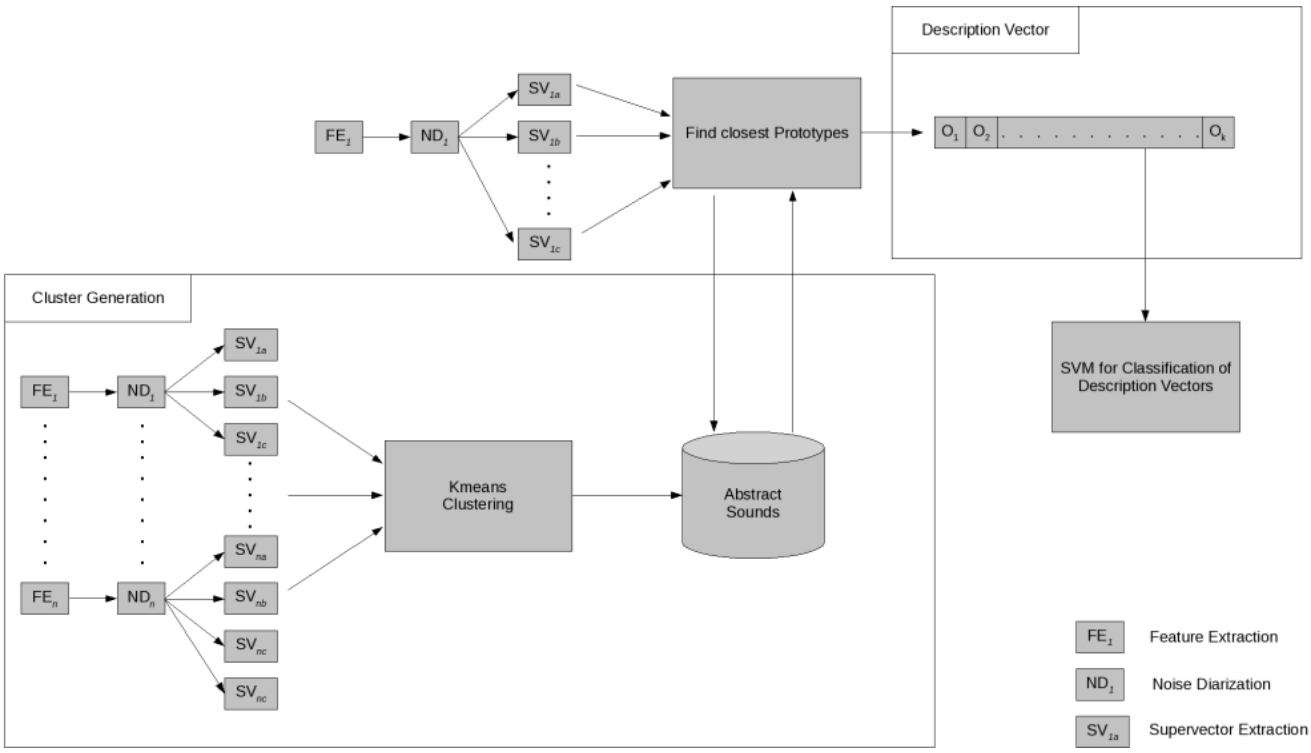

Figure 6: System overview

6.2. Experiments on MED 2010 Dataset

In order to compare our results with approaches found in the literature, using data from the NIST TRECVID MED 2010 challenge. (Jiang et al. 2010) evaluated their system on the MED 2010 dry-run validation set in terms of mean average precision (MAP). (Chadhuri et al. 2011) report their system's performance in terms of average classification accuracy on the full MED 2010 data set. We have generated diarization models based on MFCC features with a frequency range of 0-8000 KHz for the MED 2010 data. We have then clustered models for both the dry-run set and the full MED 2010 data set (evaluation set) sand used the clusters in the above described system. As the minimum segment length for diarization we have chosen 32 milliseconds based on observations made with the synthetically compiled sound set described above. In order to accommodate the change in minimum segment length we have experimented with different values for k. Table 5 shows MAP for the dry-run set with clustering using 10 and 20 iterations.

k	TF	TFIDF	logTF	logTFIDF
Kmeans with 10 iterations				
300	0.296	0.306	0.308	0.310
400	0.281	0.264	0.282	0.272

500	0.355	0.357	0.351	**0.366**
600	0.266	0.272	0.276	0.281
800	0.326	0.331	0.330	0.333
1000	0.282	0.283	0.284	0.286
1500	0.332	0.342	0.346	0.349
3000	0.289	0.294	0.307	0.314
6000	0.346	0.345	0.347	0.344
Kmeans with 20 iterations				
300	0.252	0.250	0.250	0.257
400	0.238	0.238	0.238	0.239
500	0.338	0.331	0.342	0.330
600	0.330	0.334	0.330	0.334
800	0.264	0.268	0.265	0.269
1000	0.327	0.329	0.330	0.332
1500	0.373	0.368	**0.377**	0.375
3000	0.327	0.327	0.33	0.331
6000	0.356	0.346	0.347	0.347
1500 (30 it)	0.350	0.350	0.351	0.353
300 (30 it)	0.252	0.250	0.250	0.257

Table 5: MAP for MED 2010 dry-run set

(Jiang et al. 2010) report a MAP value of 0.404 for their system. The best MAP our system achieved was 0.377 with logTF weighting and k=1500 using 20 iterations. While this value is very close to the one reported by (Jiang et al. 2010), is is still 0.027 below their result. It should be noted, however, that the system designed by (Jiang et al. 2010) is based on manual segmentation of specific sounds while our system is applicable to arbitrary domains without the need of specific manual tagging and selection of sound segments.
(Chadhuri et al. 2011) reported results in terms of average classification accuracy. When compared to the system presented by (Chadhuri et al. 2011) which is based on unsupervised training, our system outperforms their system on the 4-class discrimination task for both parameter combinations presented in their paper and for one parameter combination on the 3-class discrimination task using the full MED 2010 data set. A comprehensive comparison is given in table 6.

System	Weighting	3-class	4-class
K=300, 10 iterations	TF	67.15	65.69

	TFIDF	67.88	65.69
	logTF	66.42	67.69
	logTFIDFlogTDIDD	**68.61**	65.69
K=1500, 20 iterations	TF	65.69	92.30
	TFIDF	65.69	92.12
	logTF	67.69	**92.30**
	logTFIDFlogTDIDD	65.69	92.24
Majority Guess		35.15	90.43
64 symbols 2-gram (Chadhuri et al. 2011)		81.86	73.61
200 symbols 3-gram (Chadhuri et al. 2011)		55.63	77.08

Table 6: Average classification accuracy for MED 2010 evaluation data set.

7. Conclusion and Future Work

This paper has shown how speaker diarization can be used for audio indexing and audio based concept detection. Speaker diarization in conjunction with higher level clustering delivers abstract sound concepts that can be used for sound file indexing. These indexes can be used for classification purposes. A major advantage of the diarization based approach presented in this paper over other unsupervised approaches is that the sound concepts found can be identified and replayed to human listeners. In terms of performance a proof-of concept system build with the speaker diarization approach outperformed a state-of-the-art unsupervised approach in three out of four cases on the MED 2010 evaluation data set and achieved results nearly as good as a supervised training approach on the MED 2010 dry-run data set. This

Another benefit of the use of diarization software as a first processing step is that diarization results can be analyzed easily as shown in the section on diarization tuning. Diarization results can be analyzed both manually as well as semi-automatically using synthetically compiled sound files. The analysis results discussed in this paper have directed the choice of parameter settings for diarization that differ dramatically from those used in speaker diarization. The manual analysis results also suggest the use of non-speech related features for diarization. Future work will thus include experiments on more general compressed domain features. Other perspectives for future work are experiments with different clustering techniques as well as the use of temporal models that retain the order of sound instances in a sound file.

7. Acknowledgements

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20066. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any

Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

8. References

Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: A library for support vector machines," ACM Transactions on Intelligent Systems and Technology, vol. 2, pp. 27:1-27:27, 2011.

S. Chaudhuri, M. Harvilla, and B. Raj, "Unsupervised learning of acoustic unit descriptors for audio content representation and classification," in Interspeech, 2011.

G. Friedland, O. Vinyals, Live Speaker Identification in Conversations. Proceedings of ACM International Conference on Multimedia (ACM Multimedia 2008), Vancouver, Canada, October 2008, pp1017-1018.

J. Huang, Z. Liu, Y. Wang, Y. Chen, and E.K. Wong, "Integration of multimodal features for video scene classification based on HMM," in Multimedia Signal Processing, 1999 IEEE 3rd Workshop on, 1999, pp. 53 -58.

Yan Huang, Oriol Vinyals, Gerald Friedland, Christian Muller, Nikki Mirghafori, and Chuck Wooters, "A fast-match approach for robust, faster than real-time speaker diarization," in ASRU, 2007.

David Imseng and Gerald Friedland, "Robust speaker diarization for short speech recordings," in Proceedings of the IEEE workshop on Automatic Speech Recognition and Understanding, 12 2009, pp. 432-437.

Y.-G. Jiang, X. Zeng, G. Ye, S. Bhattacharya, D. Ellis, M. Shah, and S.-F. Chang, "Columbia-ucf trecvid2010 multimedia event detection: Combining multiple modalities, contextual concepts, and temporal matching," in NIST TRECVID Workshop, 2010.

Edda Leopold and Jorg Kindermann, "Text categorization with support vector machines. how to represent texts in input space?," Mach. Learn., vol. 46, pp. 423-444, March 2002.

M. Lan and H.-b. Low, "A comprehensive comparative study on term weighting schemes for text categorization with support vector machines," in Posters Proc. of WWW2005, 2005, pp. 1032 1033.

Michael S. Lew, Nicu Sebe, Chabane Djeraba, and Ramesh Jain, "Content-based multimedia information retrieval: State of the art and challenges," ACM Trans. Multimedia Comput. Commun. Appl., vol. 2, no. 1, pp. 1-19, 2006.

Huan Li, Lei Bao, Zan Gao, Arnold Overwijk, Wei Liu, Long fei Zhang, Shoou-I Yu, Ming yu Chen, Florian Metze, and Alexander Hauptmann, "Informedia@trecvid 2010," in Notebook for NIST's TREC Video Retrieval Evaluation 2010, 2010.

Lie Lu and A. Hanjalic, "Audio keywords discovery for textlike audio content analysis and retrieval," Multimedia, IEEE Transactions on, vol. 10, no. 1, pp. 74 -85, jan. 2008.

Mertens, R., Lei, H., Gottlieb, L., Friedland, G. & Divakaran, A. (2011) Acoustic Super Models for Large Scale Video Event Detection. ACM Multimedia 2011, International ACM Workshop on Events in Multimedia (EiMM11), Scottsdale, Arizona, USA, November 28 - December 1, 2011.

NIST TRECVid evaluation (2011). Retrieved December, 15 2011 from http://www-nlpir.nist.gov/projects/trecvid/.

Stephen Robertson, "Understanding inverse document frequency: On theoretical arguments for idf," Journal of Documentation, vol. 60, pp. 2004, 2004.

Cees G. M. Snoek and Marcel Worring, "Concept-based video retrieval," Fundamental Trends in Information Retrieval, vol. 2, no. 4, pp. 215-322, 2009.

P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, H. Meinedo, M. Bugalho, and I. Trancoso. On the use of audio events for improving video scene segmentation. In WIAMIS 2010, pages 1 -4, 2010.

HD Wactlar, T. Kanade, MA Smith, and SM Stevens, "Intelligent access to digital video: Informedia project," Computer, vol. 29, no. 5, pp. 46-52, 1996.

Chuck Wooters and Marijn Huijbregts, "Multimodal technologies for perception of humans,". The ICSI RT07s Speaker Diarization System, pp. 509-519. Springer-Verlag, Berlin, Heidelberg, 2008.