I-SED: an Interactive Sound Event Detector

Bongjun Kim
Northwestern University
Evanston, USA
bongjun@u.northwestern.edu

Bryan Pardo
Northwestern University
Evanston, USA
pardo@northwestern.edu

ABSTRACT
Tagging of sound events is essential in many research areas. However, finding sound events and labeling them within a long audio file is tedious and time-consuming. Building an automatic recognition system using machine learning techniques is often not feasible because it requires a large number of human-labeled training examples and fine tuning the model for a specific application. Fully automated labeling is also not reliable enough for all uses. We present I-SED, an interactive sound detection interface using a human-in-the-loop approach that lets a user reduce the time required to label audio that is tediously long (e.g. 20 hours) to do manually and has too few prior labeled examples (e.g. one) to train a state-of-the-art machine audio labeling system. We performed a human-subject study to validate its effectiveness and the results showed that our tool helped participants label all target sound events within a recording twice as fast as labeling them manually.

ACM Classification Keywords
H.5.2 User Interfaces: Interaction styles; H.5.5 Sound and Music ComputingSystems

Author Keywords
interactive machine learning; sound event detection; human-in-the-loop system

INTRODUCTION
Detecting sound events in recordings and giving them labels is a key technology with applications in many areas: labeling speech recordings with speaker names [18], labeling music recordings by predominant instrument [5], labeling nature recordings with the species of animals heard in the recording [13], and identifying gunshots in city recordings [20].

Even though manual annotation by human experts leads to more accurate results than automatic annotation, there are many situations where hand-labeling events in recordings is prohibitively labor intensive. For example, speech and language pathologists often wish to label sound and speech events in day-long (24 hours) recordings of an individual patient's environment. Therefore, researchers have put significant effort into developing more accurate automatic sound recognition systems. Many widely used methods for building recognition systems use supervised statistical machine learning. Examples include neural networks [10, 15], Gaussian Mixture Models (GMM) [18, 22], decision trees [12] and Support Vector Machines (SVM) [9, 17].

Making a general recognition device using these machine learning techniques typically requires a large number of labeled training examples (e.g. thousands or tens of thousands of labeled sounds). It also requires fine-tuning the model for the specific application, usually by machine learning experts. This is not feasible in the case where users do not have thousands of pre-labeled examples of the target sound. It is also not feasible when providing enough labeled examples is equivalent to solving the task manually (e.g. the user labels the entire 24-hour recording to give the machine enough training data to label the 24-hour recording). Even with lots of training data and model tuning, machine labels may not show sufficient agreement with human labels. For example, the current state of the art environmental labeling for language assessment, the LENA system [25], agrees with human annotators only 76% of the time on a four-way forced choice labeling task.

We wish to address audio labeling tasks that fall in a midground: there is too much audio to be labeled effectively by hand, yet there are too few training examples to effectively train an accurate model. Our goal is to develop an efficient way to achieve human-level accuracy with much less human effort than is typical for manual annotation. We further require the end user should be able to perform the labeling without any knowledge about machine learning or audio signal processing.

In this work, we present I-SED, an interactive sound event detector using a human-in-the-loop approach where human and machine collaborate to speed up the sound event labeling. The basic idea is to engage users to provide relevance feedback [6] to the machine labeler. We use this approach to find regions that contain similar sound events to the target sound. The user provides a few (one or two) examples to the machine. The machine's initial labeling of the audio, based on these examples, is presented to the human for validation. The human validates or modifies the machine's labels. The machine updates its labeling. This process repeats. The goal is to quickly finish the labeling task at hand. This is a fundamentally different goal compared to prior work. In prior work the interactive learning is used to train a generalized model to retrieve more relevant items or for later use on different data. To evaluate our system's effectiveness, we built a prototype interface and performed a human subject study.

IUI 2017, March 13 - 16, 2017, Limassol, Cyprus

ACM 978-1-4503-4348-0/17/03 ...$15.00.
DOI: http://dx.doi.org/10.1145/3025171.3025231

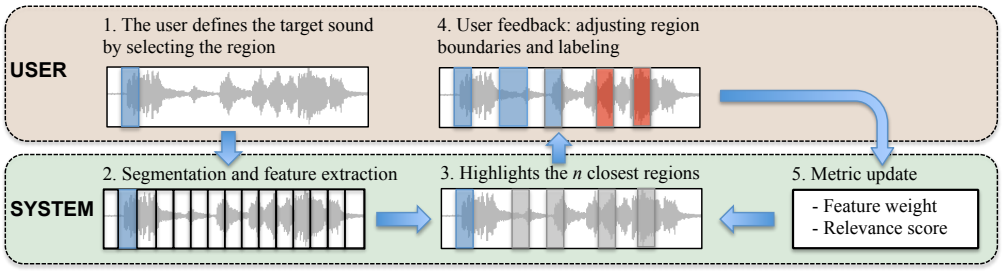

Figure 1. System overview of the interactive sound event detector

RELATED WORK

Several audio editing applications such as *Audacity* [14] and *Sonic Visualizer* [3] provide an annotation environment where a user manually selects a sub-section of an audio track and labels it. *ASAnnoatation* [2] provides low-level feature information (e.g. pitch content) to help users label, but does not suggest which high-level semantic labels (e.g. Bob's voice) to apply. It also does not allow user-defined labels.

TotalRecall [11] is a semi-automatic multimedia annotation tool. It automatically detects speech regions on an audio track (speech or non-speech) for audio segments. It helps a user to find speech sections of an audio track easily, but is hard-coded to find speech and cannot be on-the-fly re-purposed for detecting other kinds of events.

SoundsLike [7] is a tool to detect user-selected sound events in a movie. It provides similarity graph that visualizes which audio segments are similar to the user-selected segment as an aid for easier navigation. The system does not update its similarity estimates based on user relevance feedback. Therefore, if the system thinks two segments are similar and the user does not, there is no way to correct the system. Also, they did not evaluate how much the similarity graph helps the annotation process and the interface does not provide any machine prediction to speed up the labeling process.

Gulluni [8] suggested an interactive approach to label sound objects within an electro-acoustic music track. Their system does not allow a user to change boundaries of segmented regions, our system utilizes boundary adjustment of segments as user feedback to retrain a model. Their approach uses clustering techniques that require a user to listen to the audio multiple times to determine the best segmentation level. Multiple listenings can be problematic for long audio files (hours long). They also did not conduct a user study and only tested their system in simulation. In contrast, we performed human-subject study where participants actually tried our tool. Further, we implemented a web-based annotator anyone can use.

Interactive learning by users' relevance feedback has been actively researched in image retrieval. *CueFlick* [4, 1] is an image search application that allows users to create and adjust rules for concepts (e.g. portraits of people) by providing the machine with positive and negative examples. The user feedback iteratively updates the rule to obtain more accurate image search result. Their interactive approach is aimed at training the best classifier to retrieve images relevant to a query. Our goal is to completely label the audio easily and quickly. Moreover, labeling an audio track requires a different type of interface, where a user adjusts time boundaries of sound events and the updated information is used to retrain models.

Crowd-sourced human power was used in [21]. Even though crowd-sourcing annotation is a great way to collect a lot of labeled data, it is not appropriate in a situation where the audio data should be annotated by domain experts or must not be distributed in public, such as audio recordings of patients for clinical purpose.

INTERACTIVE SOUND ANNOTATION

We now describe an interactive system that lets a single user reduce the time required to label audio that is tediously long for a human (e.g. 20 hours), has target sounds that are sparse in the audio (10% or less of the audio contains the target), and has too few prior labeled examples (e.g. one) to train a state-of-the-art machine audio labeling system.

System overview

Figure 1 shows how our system works with the user to label target sound events. First, a user uploads an audio track into the system and defines the target sound (e.g. someone coughing) by selecting a region on the audio track containing an example of this sound. If the user cannot find an initial example in the audio, they may also provide a short example audio file containing an example.

The system segments the track into small regions whose length is the same as the initial example and measures features of the audio file. It then finds the n regions with features most similar to the example and returns them as candidate examples of the target sound class (e.g. coughing). The user gives feedback by labeling the candidate regions as positive or negative.

Based on the user-validated examples, the importance of features is weighted. In the new feature space, the system computes the likelihood of each unlabeled region having the target sound and again hands the user the n most relevant regions. This process of selecting candidate regions for human evaluation is repeated for a number of rounds. As more examples are labeled by the user and the features are re-weighted every round, the system's suggestion becomes more accurate.

Segmentation and feature extraction

Once a user provides the initial example to the system (e.g. a 3-second region of a bird call), the entire track is split into segments whose length is the same as the length of the initial example (e.g. 3 seconds). To measure distances between the

regions, audio features are extracted over each fixed-length segment. The length of labeled regions could vary by user feedback.

We use the first 13 MFCCs (Mel Frequency Cepstral Coefficients) as audio features. These have been used in many sound recognition tasks [16]. Each segment is split into a sequence of short frames (e.g. a frame-size of 90ms with 50% overlap between adjacent frames) and MFCCs are computed on each frame. Features extracted frame-wise are pooled over each segment (e.g. 3 seconds) using mean and variance of instantaneous and delta values. The delta values are the difference between feature values of two consecutive frames. These represent basic temporal characteristics of the feature vectors in one segment. As a result, a 52-dimnesional feature vector is built for each segment (13 MFCC averages, 13 MFCC variances, 13 MFCC average delta, 13 MFCC average delta variance) and distances between regions are measured in the feature space.

Relevance score

In each round, the system computes the relevance score of all unlabeled segments, ranks them, and presents the top n segments to the user. We apply a simple nearest neighbor (NN) approach used in [6]. The relevance score of an audio segment s is computed as:

$$Rel(s) = \frac{d(s,s_n)}{d(s,s_n)+d(s,s_p)} \quad (1)$$

where s_p and s_n are the nearest positive and nearest negative segments. Function $d(a,b)$ is the weighted Euclidean distance between two segments in the feature space. To obtain a more accurate relevance score in each round, the system re-weights features using Fisher's criterion [24]. The weight of the i^{th} feature is computed as:

$$w(i) = \frac{(avg(f_i^p) - avg(f_i^n))^2}{std(f_i^p)^2 + std(f_i^n)^2} \quad (2)$$

where f_i^p and f_i^n are vectors whose elements are i^{th} feature values of all positive and negative examples, respectively.

User relevance feedback

The system presents n regions to be labeled every round and the user listens to each region and labels them. Labeling regions plays an important role as feedback for the future rounds because the machine's suggestion for each round depends on the user feedback in the past rounds.

Users provide two kinds of feedback to the system, as shown in Figure 2. One is to apply positive or negative labels to each candidate example, which is widely used in interactive image retrieval systems [23]. The other is to adjust boundaries of the suggested region if the region does not properly cover the whole duration of a target sound event. This kind of feedback is typically not used in document or image retrieval systems, but is useful for improving retrieval of regions of audio files.

Our system automatically collects additional negative examples from the user's boundary adjustments. As shown in figure 3, for example, suppose the user changes the position of the

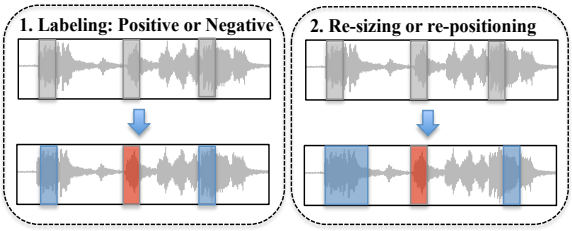

Figure 2. As feedback, a user labels regions positive(blue)/negative(red), or changes the time position and size

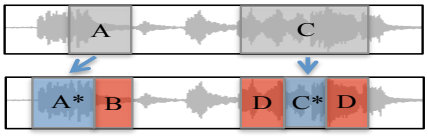

Figure 3. Regions a user listened to, but did not label are labeled as negative automatically

region (A) and labels it as positive. In this case, we can obtain not only one positive example, but also one negative example which is the region (B) that the user did not select, but listened to. In the same way, adjusting boundaries of the region (C) generates negative examples (D). This automatic negative labeling is beneficial in two ways: 1) A user implicitly labels more regions, speeding interaction, and 2) Since our system presents the most relevant examples to a user every round, the pool of labeled examples tends to skew towards positive, which could make measuring relevance score problematic. Therefore, adding negative examples automatically helps in computing more accurate relevance scores of unlabeled examples.

IMPLEMENTATION

Figure 4 shows the main workspace of the proposed interface. It consists of three main sections: *Navigation Map*, *Annotation Track*, and *Region Selection*. The *Navigation Map* displays a waveform of an entire track and the currently labeled regions so that a user navigates and listens to them easily. The *Annotation Track* is a zoomed-in version of *Navigation Map*. A user can select regions or change their boundaries by mouse clicking and dragging. The *Region Selection* displays the top n candidate regions identified by the machine. The user can listen to presented regions by clicking the items in the list and label it by clicking <*positive*> or <*negative*> button. Once the user labels all of them in that round, the user clicks on <*find similar regions*> button to give system feedback and obtains a new set of candidate regions from the system. Readers can watch the demo video and try the system out at `http://www.bongjunkim.com/ised`.

EVALUATION

We conducted a user study to verify whether our interactive annotation system lets a user detect and label target sounds in the given audio track faster than with manual annotation. For manual annotation, we provided the identical interface to the interactive one except for removal of systems recommendations. We recruited 20 potential users of our tool, including people who study speech and language development and researchers in machine learning and audio.

Figure 4. Screenshot of the interactive sound event detector

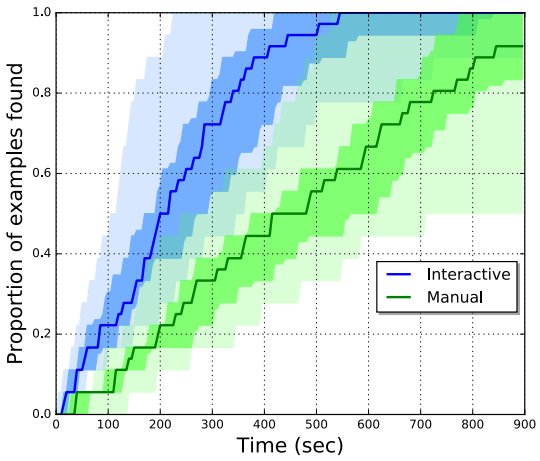

Figure 5. The proportion of examples found over time (quantized every 5 seconds) using two different interfaces, our proposed interactive system and a manual annotator. Here, N = 20, as each of 20 participants tried both interfaces in a session. Lines indicate medians, and dark and light bands of each color show 75th and 25th percentile.

Dataset

We used the dataset from the Detection and Classification of Acoustic Scenes and Events (DCASE) challenge. [19]. To generate testing tracks for this experiments, we chose files used for the Office Synthetic (OS) Event Detection Task of the DCASE 2016 challenge.[1] They are two minute-long mono recordings of sequences containing artificially concatenating overlapping acoustic events in an office environment (e.g. coughing, drawer, door knock, speech, etc.). We created each 12 minute-long audio track by concatenating six short tracks in the dataset to increase the length of an audio file to be over the 10-minute length we anticipate as the minimal length where someone might wish to speed search. Each 12-minute track has 11 different sound classes with 18 examples of each class in the track and all events are randomly distributed over the track. The total time proportion that each class takes up in a track is roughly 4% of the entire length of the track.

Task procedure

Each subject participated in one session. Each session consists of two annotation tasks: one task with the manual annotation interface and the other with the proposed interactive annotation interface. Prior to each task, users were given a 4-minute training session to learn the interface. For each task, the participant was given 15 minutes to find as many target sounds as they could within a 12-minute recording using one of the two interfaces: manual annotator or our interactive detector.

We created two 12 minute-long audio tracks (one for each task) by randomly reordering sound events in the dataset. A unique recording is assigned to each task to control the learning effect. We had participants search for two sonically different target sounds: knocks and speech. The presentation order of both interfaces and audio tasks were balanced designed for the unbiased result. The 20 participants were divided into 4 groups so that task and interface order was balanced.

[1] http://www.cs.tut.fi/sgn/arg/dcase2016/task-sound-event-detection-in-synthetic-audio

Results

Figure 5 shows the proportion of the target sound events detected by the 20 participants as a function of time they spent. We considered a sound event correctly detected if the user-labeled region overlaps sufficiently with its ground truth (tolerance for the onset and offset: 1 second). Participants spent an average of 517 seconds labeling all target sound events using the interactive detector. It is about 15 rounds per user. The mean times that participants spent finding 80% of the sound events are 740 seconds for the manual annotator and 347 seconds for the interactive detector (Wilcoxon signed-rank test: $p<0.05$). We can conclude that the interactive detector helped participants find sound events roughly twice as fast as the manual annotator. We also compared the interactive detector to a fully automated baseline system which is the initial ordering of the audio segments provided by the system, prior to any user feedback. While participants evaluated about 15% of the total duration of the audio to find all target sounds, the baseline finds all target sounds when it returns top 33% of it.

CONCLUSIONS

We presented a new system for sound event detection and annotation using interactive learning focusing on helping the user label as many target sound events as possible every round. The experiment showed that the proposed interface lets users find sparsely-distributed target sounds roughly twice as fast as manually labeling the target sounds. The effectiveness of the proposed approach depends on the retrieval accuracy of the machine and the user interaction with the system. We will explore alternate retrieval techniques and a new interaction design in the context of interactive learning. When to stop labeling is also an important issue. Developing a systematic stopping criterion is one area for future work.

ACKNOWLEDGMENTS

This work was supported by NSF Grant 1617497.

REFERENCES

1. Saleema Amershi, James Fogarty, Ashish Kapoor, and Desney Tan. 2010. Examining multiple potential models in end-user interactive concept learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1357–1360.
2. Niels Bogaards, Chunghsin Yeh, and Juan José Burred. 2008. Introducing ASAnnotation: a tool for sound analysis and annotation. In *ICMC*. 1–1.
3. Chris Cannam, Christian Landone, Mark B Sandler, and Juan Pablo Bello. 2006. The Sonic Visualiser: A Visualisation Platform for Semantic Descriptors from Musical Signals.. In *ISMIR*. 324–327.
4. James Fogarty, Desney Tan, Ashish Kapoor, and Simon Winder. 2008. CueFlik: interactive concept learning in image search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 29–38.
5. Ferdinand Fuhrmann, Martín Haro, and Perfecto Herrera. 2009. Scalability, Generality and Temporal Aspects in Automatic Recognition of Predominant Musical Instruments in Polyphonic Music.. In *ISMIR*.
6. Giorgio Giacinto. 2007. A nearest-neighbor approach to relevance feedback in content based image retrieval. In *Proceedings of the 6th ACM international conference on Image and video retrieval*. ACM, 456–463.
7. Jorge Gomes, Teresa Chambel, and Thibault Langlois. 2013. SoundsLike: movies soundtrack browsing and labeling based on relevance feedback and gamification. In *Proceedings of the 11th european conference on Interactive TV and video*. ACM, 59–62.
8. Sébastien Gulluni, Slim Essid, Olivier Buisson, and Gaël Richard. 2011. An Interactive System for Electro-Acoustic Music Analysis.. In *ISMIR*. 145–150.
9. Guodong Guo and Stan Z Li. 2003. Content-based audio classification and retrieval by support vector machines. *Neural Networks, IEEE Transactions on* 14, 1 (2003), 209–215.
10. Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, and others. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE* 29, 6 (2012), 82–97.
11. Rony Kubat, Philip DeCamp, Brandon Roy, and Deb Roy. 2007. Totalrecall: visualization and semi-automatic annotation of very large audio-visual corpora.. In *ICMI*, Vol. 7. 208–215.
12. Yizhar Lavner and Dima Ruinskiy. 2009. A decision-tree-based algorithm for speech/music classification and segmentation. *EURASIP Journal on Audio, Speech, and Music Processing* 2009 (2009), 2.
13. Chang-Hsing Lee, Chin-Chuan Han, and Ching-Chien Chuang. 2008. Automatic classification of bird species from their sounds using two-dimensional cepstral coefficients. *Audio, Speech, and Language Processing, IEEE Transactions on* 16, 8 (2008), 1541–1550.
14. Beinan Li, John Ashley Burgoyne, and Ichiro Fujinaga. 2006. Extending Audacity for Audio Annotation.. In *ISMIR*. 379–380.
15. Giambattista Parascandolo, Heikki Huttunen, and Tuomas Virtanen. 2016. Recurrent neural networks for polyphonic sound event detection in real life recordings. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6440–6444.
16. Geoffroy Peeters. 2004. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. *Technical report, IRCAM* (2004).
17. Jose Portelo, Miguel Bugalho, Isabel Trancoso, Joao Neto, Alberto Abad, and Antonio Serralheiro. 2009. Non-speech audio event detection. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 1973–1976.
18. Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. 2000. Speaker verification using adapted Gaussian mixture models. *Digital signal processing* 10, 1 (2000), 19–41.
19. Dan Stowell, Dimitrios Giannoulis, Emmanouil Benetos, Mathieu Lagrange, and Mark D Plumbley. 2015. Detection and classification of acoustic scenes and events. *Multimedia, IEEE Transactions on* 17, 10 (2015), 1733–1746.
20. Giuseppe Valenzise, Luigi Gerosa, Marco Tagliasacchi, Fabio Antonacci, and Augusto Sarti. 2007. Scream and gunshot detection and localization for audio-surveillance systems. In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*. IEEE, 21–26.
21. Sudheendra Vijayanarasimhan and Kristen Grauman. 2014. Large-scale live active learning: Training object detectors with crawled data and crowds. *International Journal of Computer Vision* 108, 1-2 (2014), 97–114.
22. L Vuegen, BVD Broeck, P Karsmakers, JF Gemmeke, B Vanrumste, and HV Hamme. 2013. An MFCC-GMM approach for event detection and classification. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 1–3.
23. Xiang-Yang Wang, Bei-Bei Zhang, and Hong-Ying Yang. 2013. Active SVM-based relevance feedback using multiple classifiers ensemble and features reweighting. *Engineering Applications of Artificial Intelligence* 26, 1 (2013), 368–381.
24. Emin Wu and Aidong Zhang. 2002. A feature re-weighting approach for relevance feedback in image retrieval. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, Vol. 2. IEEE, II–581.
25. Dongxin Xu, Umit Yapanel, and Sharmi Gray. 2009. *Reliability of the LENATM Language Environment Analysis System in young children's natural home environment*. Technical Report. LENA Foundation Technical Report LTR-05-02). Retrieved from http://www.lenafoundation.org/TechReport. aspx/Reliability/LTR-05-2.