INTERSPEECH 2009 BRIGHTON

Automatic Detection of Audio Advertisements

I. Dan Melamed and Yeon-Jun Kim

AT&T Labs-Research, 180 Park Avenue, Florham Park, NJ, 07932, USA
{{lastname},yjkim}@research.att.com

Abstract

Quality control analysts in customer service call centers often search for keywords in call transcripts. Their searches can return an overwhelming number of false positives when the search terms also appear in advertisements that customers hear while they are on hold. This paper presents new methods for detecting advertisements in audio data, so that they can be filtered out. In order to be usable in real-world applications, our methods are designed to minimize human intervention after deployment. Even so, they are much more accurate than a baseline HMM method.

1. Introduction

The customer service call centers of many companies record their calls for quality control purposes. As part of their efforts, quality analysts use voice search systems to search these recordings [1]. A typical architecture for such a system is illustrated in Figure 1. When a service call is recorded, it is stored on a media server for future playback. It is also transcribed by automatic speech recognition (ASR) software (e.g. [2]). The transcript is then indexed for searching. Later, a quality analyst performs searches on this index, to retrieve a list of calls that satisfy various criteria, such as calls whose transcripts include certain words or phrases. The analyst can then choose to retrieve some of the calls from the media server and listen to them.

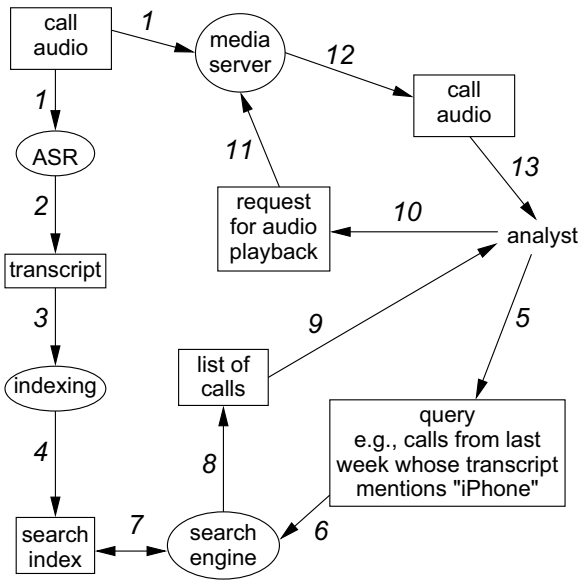

Figure 1: Architecture of a typical system for call center quality control. Numbers show the typical sequence of events.

Unfortunately, in addition to the speech of the customers and the customer service agents, call recordings often contain advertisements that customers hear while they are on hold. When an analyst searches for product or service names, the majority of their search results can consist of calls where the search terms are mentioned only in ads. To find calls where the search terms are spoken by the customer or the agent, the analyst has no choice but to waste a lot of time listening to irrelevant calls. The analysts' efforts can be greatly facilitated by a system for filtering ads from call transcripts, so that the ads do not pollute search results. Filtering ads would be easy if the system used to play ads during calls made a record of each time it starts or stops. Alas, in the real world, such systems are typically unmodifiable black boxes, and no timestamps can be extracted from them. Thus, the need arises for automatic detection of advertisements using nothing but the audio stream itself.

There has been some previous work to separate speaker segments in environments where the number of speakers is known [3] or where it is reasonable to assume a minimum duration for each speaker segment and/or clear pauses between speaker segments [4]. These heuristics are unsuitable for our audio recordings, where there can be many speakers and the dialogues involve frequent barge-ins. In any case, the ad detection problem is different from the problem of speaker segmentation. An ad, or a contiguous sequence of ads, might involve multiple voices. Likewise, non-ad segments of a given call recording might involve several customers and/or several customer service agents. To detect ads, it does not help to segment call recordings into different speakers within ad segments and within non-ad segments. Even if perfect speaker segmentation were possible for call recordings, it would still leave open the question of which speakers are reading ads.

To be feasible for real-world applications, an ad detection method should not require frequent human intervention. This criterion rules out ad detection based on text-independent speaker recognition [5]. A speaker recognition system might be trained to detect a particular set of voices used in a particular set of ads, but if new ads use a different set of voices, then the system would fail to detect them. This criterion also rules out methods that try to detect previous ads based on their transcripts and/or how they sound. For example, the techniques of [6, 7] involve aligning the input to a known reference. However, it is impractical for us to keep track of when new ads come out, let alone create a new reference for each one.

In this paper, we compare five new methods for detecting ads in call recordings. Each method relies on characteristics that are common to audio advertisements in general, and not to particular advertisements or particular speakers. Therefore, each method needs to be trained only once. Two HMM methods serve as simplistic baselines. The third method is based on our observation that the intonation of recorded advertisements is very different from that of ordinary speech. The intonation

10.21437/Interspeech.2009-459

of ads often involves long segments of emphatic stresses, which show up as sharp up slopes and down slopes in the F0 contour. [8] used intonation to separate speakers and languages. They modeled a sequence of symbolic patterns to focus on language differences, whereas we use three numeric parameters to distinguish between voices that are all speaking American English. The fourth method is based on the observation that, even though ads can change over time, every ad appears in many calls. Therefore, ads can be detected as frequently occurring word sequences in ASR output. The fifth and best method is a combination of the third and fourth. It achieves an accuracy of 92% on held-out test data, which is good enough for use in real-world applications.

2. Data

The data used in this study consisted of recorded customer service calls, whose duration ranged from 5 to 20 minutes. The audio quality of the recordings was very poor: 4-bit ADPCM sampled at 6kHz. Five such calls were drawn uniformly at random for each of 20 consecutive business days, for a total of 100 calls. A colleague who is not one of the authors annotated the advertisements in each call using the Praat audio annotation software.[1] We also ran our in-house large-vocabulary ASR system [2] over these call recordings.

The ad segments varied a great deal in their duration and the number of words that they contain. The standard deviation of ad segment duration was 11.3 seconds. The standard deviation of the number of words per ad segment was 372. With function words removed, as explained below, the standard deviation was 188.

3. Ad Detection Methods

3.1. Two-state HMM

A relatively simple way to detect ads, independently of what the ads are about, is to use a two-state ergodic Hidden Markov Model that ranges over acoustic features. One state of the HMM represents ads and the other state represents everything else. In principle, any part of the training data can be input to the procedure for estimating the HMM's emission probabilities. In our experiments we used F0, F0', F0", and the standard 39 MFCCs from 100ms frames. In a pilot experiment this frame length seemed to produce more stable results than the more common 10ms.

3.2. Three-state HMM

We hypothesized that the two-state HMM might be confounded by frames of silence. So we also built a three-state HMM, which was just like the two-state HMM, but with an additional state to represent silence.

3.3. Pitch Dynamics

In an effort to increase the listener's attention, voices in audio ads tend to vary their pitch more rapidly than voices in other kinds of speech. Figure 2 compares the distributions of F0 slopes for ads vs. all other speech in our training data. We exploited this distinguishing characteristic in a novel ad detection method.

[1] `http://www.fon.hum.uva.nl/praat`

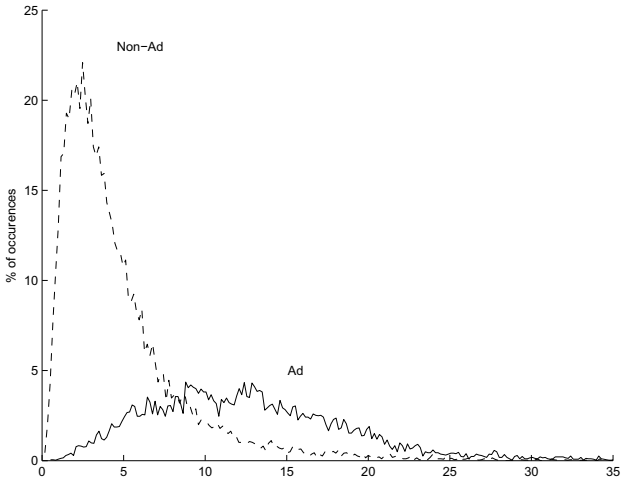

Figure 2: Distributions of pitch dynamics for speech in ads vs. all other speech. The x-axis shows the maximum change per second in F0, sampled over 10ms frames.

The method requires 3 parameters to be optimized on training data:

- the minimum pitch variance v_{min}
- the minimum gap length g_{min}
- the minimum ad length l_{min}

It then uses these parameters in the following procedure:

1. Measure the fundamental frequency (F0) in every 10ms frame of the call recording. Although the F0 contour is generally difficult to model, it is relatively stable in the face of background noise, as long as the voice of interest remains the dominant signal. For example, many of the advertisements in our data were accompanied by soft background music. In our work, we used the auto-correlation based pitch extractor get_$F0$ from the ESPS/Waves toolkit.[2]
2. Filter out F0 values that are outside the typical range for human voices. Such values can arise as artifacts of the pitch extraction algorithm.
3. Compute the change in the fundamental frequency (F0') between every two adjacent frames.
4. Find monotonic sequences of F0' values that are longer than 50ms.
5. Partition the call into snippets of 1 second each[3] and, in each snippet S, compute the following measure of pitch variance:

$$v_S = \max_{t \in S, n > 5} \left| \frac{\sum_{i=0}^{n} F0'(t+i)}{n} \right| \quad (1)$$

where t ranges over 10ms time frames in the snippet. In other words, the pitch variance of each 1-second snippet is measured as the maximum absolute slope of F0 values spanning at least 50ms. Every snippet S where $v_S > v_{min}$ is labeled as an ad snippet; the remaining snippets are labeled as non-ad snippets.

[2] `http://www.speech.kth.se/software/\#esps`

[3] Using snippets of a different length and/or overlapping snippets might improve accuracy. We leave these possibilities to future work.

6. Fill in the gaps. If two ad snippets i and k are less than g_{min} seconds apart, then label every snippet j, where $i < j < k$, as an ad snippet.
7. Impose minimum ad length. For every contiguous sequence of ad snippets $(i, \ldots, j)$, where neither $i-1$ nor $j+1$ are ad snippets, if $j - i < l_{min}$, then relabel snippets $i, \ldots, j$ as non-ad snippets.
8. Output the snippets labeled as ads.

3.4. Word n-Grams from ASR Output

A given ad is likely to appear in many call recordings. Therefore, if we consider various word sequences of length n (henceforth, n-grams) that appear in a collection of call recordings, then the n-grams that appear in ads will be much more frequent than most other n-grams. This heuristic is far from foolproof, because the ad segments in our call recordings can start and/or stop in the middle of an ad. Also, noise in the recordings can cause the ASR system to produce different outputs for different instances of the same ad. So, instead of trying to detect whole ads at a time, our method uses sequences of short overlapping n-grams.

The method involves the following 3 parameters:

- the n-gram size s
- the minimum n-gram frequency f_{min}
- the minimum word gap length w_{min}

At training time, we compute the frequencies of all n-grams, for suitable values of n, in a corpus of call transcripts.[4] Then we optimize the three parameters using the test procedure and a suitable objective function (described in Section 4).

The test procedure is as follows for each call recording:

1. Run ASR over the recording to produce a transcript.
2. Find all substrings A of the transcript, such that $|A| \geq s$ and every s-gram $a \in A$ has a frequency of at least f_{min} in the frequency tables.
3. For every pair of substrings found in the previous step, if they are separated by less than w_{min} words in the transcript, then combine them and their intervening words into one substring.
4. Output every substring found in the previous step as an ad.

3.5. Combined Method

There are some very frequent n-grams that do not come from ads, such as "Your approximate wait time is... Thank you for calling... and how are you today?" However, the intonation of frequent n-grams that do not come from ads is unlikely to exhibit much pitch variance. To raise the precision of the n-gram method, we combined it with the pitch dynamics method. Our ASR system output a timestamp for each word that it recognized. We used these timestamps to match up the positions of ads hypothesized by the n-gram and pitch dynamics methods. The combined method output ad segments hypothesized by the n-gram method that overlapped at least partially with some ad segment hypothesized by the pitch dynamics method.

[4] In practice, this corpus should be regularly updated to include the transcripts of the latest ads. (Such updates can be done without human intervention.) However, in order for the optimal f_{min} to be stable, the size of the corpus used to compute the frequency tables must remain fairly constant.

4. Experiments

4.1. Objective Function

A simple way to define ad segments and non-ad segments is as time slices of an audio stream. However, for the voice search application, we don't care about the parts of a segment that contain no speech. Moreover, since the goal is to avoid false hits during keyword searches, we don't care about the parts of speech that are unlikely to be search terms. Therefore, we deleted from the call transcripts all instances of 283 English function words (such as "the" and "who") and filler words (such as "um"). Then, instead of comparing segments of audio, we evaluated our ad detection methods in terms of how well they filter out the content words that appear in the ad segments of these content-word-only call transcripts. More specifically, each word in each transcript was annotated with the call that it came from and with its position in that call's transcript. Each ad in each call in our data was mapped to a set of these annotated words. The ad segments hypothesized by each of our five ad detection methods were also mapped to the same representation. The hypothesized and correct sets of annotated words were then compared using the standard measures of precision, recall, and their harmonic mean, also known as the F_1 measure.

4.2. Experimental Design

We used 5-fold cross-validation to evaluate each of our 5 ad detection methods. Each "fold" used a different 80/20 split into training and test sets, so that each of our 100 annotated calls appeared in a test set exactly once. At test time, the annotated word sets for all 20 test calls were pooled into one set before computing the evaluation measures, so that the result would be a micro-average.

The boundaries between ads and non-ads in our training data allowed us to compute the parameters of our two-state HMM directly, without re-estimation. To estimate the silence parameters for the three-state HMM, we applied a standard voice activity detection algorithm [9] to our training data, and then proceeded with standard maximum likelihood estimation. At test time, both HMMs were decoded using the Viterbi algorithm [10]. We used a grid search over plausible parameter values to optimize the parameters of the pitch dynamics, n-gram, and combined methods. The six parameters of the combined method were optimized together, independently of their optimization for each of the component methods.

4.3. Results

Table 1 shows the mean precision, recall, and F_1 measures for all five methods. The difference between each pair of different means is statistically significant at $p = 0.01$ using the t-test for paired samples.

Pitch dynamics turned out to be a surprisingly good way to detect ads, even on its own. The n-gram method is even more reliable. Combining these two sources of information yields a method whose error rate is 81% lower than that of the best HMM baseline.

5. Conclusion

We have presented several new methods for detecting advertisements in recordings of customer service calls. The accuracy of our best method is much higher than that of a baseline HMM method. Indeed, it is sufficiently high for daily use in our deployed voice search application.

method	precision	recall	F_1
2-state ergodic HMM	.38	.90	.54
3-state ergodic HMM	.42	.89	.57
pitch dynamics	.75	**.93**	.83
n-gram	.85	**.93**	.89
combined	**.92**	**.93**	**.92**

Table 1: Mean scores from 5-fold cross-validation. All differences are statistically significant at $p = 0.01$ using the t-test for paired samples.

In future work, we hope to find acoustic features other than the F0 contour that might be indicative of ads. We might also incorporate the output of a speaker segmentation algorithm as an additional information source, though probably not as a hard constraint. Finally, we would like to use our method to facilitate the analysis of other kinds of audio data that involve advertisements, such as radio and television programs.

6. Acknowledgments

Thanks to Barbara Hollister for annotating our data, and to the anonymous reviewers for their insightful comments.

7. References

[1] G. Zweig, O. Siohan, G. Saon, B. Ramabhadran, D. Povey, L. Mangu, and B. Kingsbury, "Automated quality monitoring for call centers using speech and NLP technologies," in *HLT-NAACL*, 2006.

[2] V. Goffin, C. Allauzen, E. Bocchieri, D. Hakkani-Tur, A. Ljolje, S. Parthasarathy, M. Rahim, G. Riccardi, and M. Saraclar, "The AT&T WATSON speech recognizer," in *ICASSP*, 2005.

[3] A. Rosenberg, Allen Gorin, Zhu Liu, and S. Parthasarathy, "Unsupervised speaker segmentation of telephone conversation," in *ICSLP*, 2002.

[4] Z. Liu and M. Saraclar, "Speaker segmentation and adaptation for speech recognition on multiple-speaker audio conference data," in *ICME*, 2007.

[5] A. Adami and H. Hermansky, "Segmentation of speech for speaker and language recognition," in *Eurospeech*, 2003.

[6] U. Turk and F. Schiel, "Speaker verification based on the german veridat database," in *Eurospeech*, 2003.

[7] M. Covell, S. Baluja, and M. Fink, "Advertisement detection and replacement using acoustic and visual repetition," in *Multimedia Signal Processing*, 2006.

[8] A. Adami, R. Mihaescu, D. Reynolds, and J. Godfrey, "Modeling prosodic dynamics for speaker recognition," in *ICASSP*, 2003.

[9] D. K. Freeman, "The voice activity detector for the pan-european digital cellular mobile telephone service," in *ICASSP*, 1989.

[10] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Transactions on Information Theory*, vol. IT-13, no. 4, pp. 260–269, Apr. 1967.