Using Speaker ID to Discover Repeat Callers of a Spoken Dialog System

Andrew Fandrianto, Brian Langner, Alan W Black

Language Technologies Institute
Carnegie Mellon University, Pittsburgh, USA
afandria@andrew.cmu.edu, {blangner,awb}@cs.cmu.edu

Abstract

This paper describes using speaker ID techniques to identify repeat callers in a spoken dialog system, using only acoustic features. Often it is useful to know if a dialog user is a novice or is experienced, and it can be the case that identifying data such as Caller ID is either unreliable or unavailable. Our approach attempts to remedy this by determining user identity in a dialog session using the acoustic information in the dialog. We optimize the audio content of each call by removing artifacts not relevant to modeling speech. This technique is applied to finding consecutive callers and creating unique user identities over all calls over a larger time frame, with the aim of tuning or adapting the dialog system based on the user identity. Our results show that the technique is effective in recognizing consecutive callers and in identifying a unique user identities in a large set of calls.

Index Terms: spoken dialog, speaker ID

1. Introduction

If callers could be reliably identified in a spoken dialog system it opens the possibility of modifying the system toward their requirements. This could be at the acoustic adaptation level, or much higher levels, such as automatically taking into account user preferences or other similar things. It is often felt that Caller ID, where the telephone of the calling party is encoded and sent between the first and second ring should be sufficient. However it is unfortunately quite common in a complex real system that the information is lost or unavailable due to legacy equipment or call forwarding. We therefore wished to investigate if we could adequately compensate Caller ID with acoustic based Speaker ID technology.

Speaker ID (also referred to as Speaker Recognition) [1] is a basic speech technology that allows identification of common speakers within a large group of users. Although it can be used for security purposes, it is often used to identify speakers to allow for better acoustic model adaptation in speech recognition systems. Speaker-ID technique range in complexity based in accuracy and computational requirements. Here we use one of the simplests forms. As we expect users to be calling most often on similar lines (their own cell phone), and speaking very similar requests (bus information), we rely on a basic Gaussian Mixture Model to try to cluster speakers.

As we actually do not have Caller ID system for the Let's Go! [2] spoken dialog system (the call forwarding setup cannot support this), we have to rely solely on Speaker ID techniques to identify calls from the same speaker. The goal of this work is to identify repeat users, using only the acoustic information from the dialog session.

Assigning IDs to determine veteran callers can be potentially useful in personalizing the dialog system. Each individual user has a set of calling patterns, pronunciation styles, and query habits. Being able to have pre-session context in which the caller is speaking can allow for on-the-fly tuning of the ASR and an expectation of which bus schedules to recommend. The long-term ideal would be to identify a user immediately as he calls, greeting him as an old friend, and then providing an instantaneous answer about his typical bus schedule.

Identifying consecutive callers may also yield useful results. Callers who run into a wall while speaking with Let's Go! often call back instead of asking the dialog system to restart. This leads to a high probability of the caller in the session following an "unsuccessful" session being the same caller. This can allow the system to learn from its mistakes. Modifying the ASR and dialog system to discount incorrect elements in the previous session could help both their outputs be more accurate.

2. Approach

2.1. Method

Our approach analyzes previously collected dialogs in an offline fashion. We assume that only a single person is participating in each dialog session. The goal is to determine which sessions have the same user identity. As such, a specific user identity is mapped to a group of acoustically related calls. We believe clustering similar calls will allow for Caller ID functionality without having actual Caller ID information. We use a GMM to model user identities, and MFCCs to model the acoustic characteristics of a session.

2.2. Initial Testing

For our initial tests, we used one month of dialog sessions from the Let's Go! system. For all sessions, we attempt to classify the session in two different ways. First, we determine if the user identity in this session has been found in any other sessions, and *automatically label the session user identity*. Secondly, we also check whether this session's identity cluster matches that of the previous session: we *find consecutive callers*. Users of this Let's Go! system are known to call back immediately for various reasons, so this happens reasonably frequently in the data.

2.2.1. Consecutive Callers

To determine if a pair of consecutive sessions has the same identity, we take each pair of adjacent sessions S_i and S_{i+1} and measure the likelihood L of seeing i given $i + 1$, denoted by $L(S_i|S_{i+1})$. If L is above a certain threshold – based on a fraction of $L(S_i|S_i)$ – the identity of S_{i+1} is considered the same as caller in S_i. The results of this approach were generally good, showing about 80% precision, as estimated by randomly sam-

Copyright © 2011 ISCA

pling 20 sessions of the approximately 200 positive matches.

2.2.2. *Labeling User Identity for Sessions*

We also used our approach to label each session with an identifier representing a unique user. We used a loose grouping to form identity clusters within a single month of dialog sessions. To create clusters, each session is assigned to a singleton cluster. Any two sessions S_i, S_j within their corresponding clusters ($S_i \in C_i, S_j \in C_j$) must be mutually compatible; that is, both $L(S_i|S_j) > T_i$ and $L(S_j|S_i) > T_j$ must be true. T_i and T_j are thresholds for each session, such that $T_i = avg_m(L(S_m|S_i)) + 4 \times sd_m(L(S_m|S_i))$ where m iterates over all calls This mutual compatibility notion attempts to capture that user identity is symmetric between calls. We did not use a single threshold value for all sessions because the likelihood that two sessions have the same caller is dependent on the particular sessions and is not necessarily comparable in all cases. The next logical step was to use the the distribution of likelihoods for any given call to decide whether a set of calls are matched or not.

If this mutual compatibility condition is satisfied, the two clusters are merged into a single cluster. In other words, membership to the cluster C_i for a session S only requires mutual compatibility between S and any session already in C_i. The justification for using a "loose" grouping that requires only a single match, as opposed to matching several or all sessions within the cluster, is that actual user speech can vary gradually over time. A single match requirement will correctly group such user identities together, while a stricter match would not. Our tests comparing these methods showed the single-link approach worked better in practice, especially amongst smaller cluster sizes. However, there is a tradeoff: larger clusters are more likely to contain more than one identity, meaning these clusters will not necessarily correspond to a single unique user.

We have investigated increasing the strictness of clusters by requiring more links proportional to cluster size. Specifically, we have tried requiring a match to have $\geq \frac{n}{2}$ mutual compatibilities for a cluster with n members.

2.2.3. *Improving the Likelihood Accuracy*

Since the success of this approach is dependent on having an accurate value for L, we tried to improve the likelihood measure in several ways. First, we tried using a background model of the session audio in an attempt to help discriminate more distinguishing signals unique to a specific identity. Here, we use "background model" to refer to a representation of the acoustics of the average user population. We hand selected a set of 10 sessions, each with different identities, that seem to capture the wide variety of user characteristics. It is important not to have any unusual features such as loud background noises or any other irregularities, since their inclusion may lead to deemphasizing the desired signal type. We use these sessions to produce a background cluster B. We then compute a modified likelihood value, $L_{new} = L(C_i|C_j) - L(B|C_j)$.

We also observed that audio segments within a dialog session had significant silence regions at the beginning – likely due to users pausing before taking their turn in the dialog – and also at the end, which is primarily due to the endpointer needing to detect some amount of silence before starting a new segment. Because most dialog turn segments are short – on the order of a few seconds – lengthy silences will end up comprising a signficant amount of audio time. Since silence often matches well with silence, this has a negative influence on our ability to detect user identities. We pruned the leading and trailing silences in each audio segment to a maximum silence length of 0.2 seconds, using the silence pruner from the Festival [3] synthesis system. Total audio size dropped 40% after applying this, and this seems to directly correlate to models being tighter.

Additionally, we attempted to use F0 directly as a distinguishing feature between user identities. F0 should be able to provide more acoustic evidence for two identities being the same person or not. However, since the audio is collected from real users via the telephone system and thus is of variable quality, we were not able to reliably extract F0 information. Examining the results of F0 extraction, we found the reported frequency to be less than 50% accurate. Using F0 values of that inaccuracy resulted in poorer performance, so we did not use F0 to modify the likelihood values.

3. Finding Frequent Users

3.1. Experiment

We used one year of dialogs (approximately 15000 sessions) from Let's Go!, looking to find repeat users over a reasonably long time frame using the same optimized approach as in the initial tests. The presence of DTMF in the calls was a new challenge; previously when analyzing consecutive calls it was not an issue as subsequent users who both use DTMF have a high chance of being the same person. However, when considering all calls, many unrelated sessions would be grouped together based solely on identifying DTMF. Though examining the differences between DTMF and non-DTMF users may be useful, this work is attempting to identify unique individuals, not finding multiple people who use DTMF. Thus, we chose to omit DTMF from the acoustic features used by the identity classifier.

We trained a model to detect and ignore audio samples with DTMF, to keep them from being labeled as a speaker identity. First, a rudimentary DTMF model D_1 was created with generated DTMF waveforms. D_1 was decoded on all caller turns. Top likelihoods (cut off by some threshold) of $L(D_1|Turn_i)$ were taken to form another model D_2. Creating and using D_2 as the secondary DTMF model allowed a broadening of the matches, specializing specifically for the set of data operated on. Top likelihoods (again cut off by some threshold) of $L(D_2|Turn_i)$ were removed from acoustic consideration. Though we used this for DTMF sounds, this approach is general enough to be used for any non-speaker generated sound that should be ignored in the identity labeling.

Creating a complete graph between all sessions for increasingly larger sets gets exponentially harder, as it is necessary to compute $\frac{n(n-1)}{2}$ mutual likelihoods between each n sessions. To avoid computing nearly 90 million edges in this graph, we split the year into its 12 months. We then created a complete graph on each month, extracted clusters greater than 3 elements from each month, and compared these month-level clusters with each other. The merge conditions for two month-level clusters M_i, M_j require mutual compatibility with respect to all member sessions. The metric is similar to the intra-month matching, except the model and thresholds are trained on all sessions in M_i, M_j. There were few clusters which contained sessions spanning multiple months. Lowering thresholds allowed more multiple month clusters, but also resulted in combining clusters that represented competely different callers.

We attempted to improve the multi-month clusters by training the model and thresholds on only the most canonical sessions of each month-level cluster. We defined "most canonical" as those sessions in a subset of the cluster which have the high-

Cluster Size	# Clusters	% Single Identity
<5 Sessions	106	60%
5-9 Sessions	26	45%
10+ Sessions	8	25%

Table 1: Cluster quality as a function of cluster size.

est likelihood of observing the cluster given the session. Though we felt this would produce better matches for combining clusters across months, we found no significant difference in cluster quality using several different sizes of the training subset, from one third to one half the cluster size.

3.2. Results

We found 140 clusters of at least 3 sessions. These clusters represent caller identities; ideally, each cluster should correspond to exactly one real identity. Thus, one way to measure cluster quality is to determine how many actual identities are contained within it. We had several people listen to the dialog sessions in the clusters, and judge how many different people they heard. It should be noted that this task is quite challenging for most people, especially as the cluster size becomes large. Further, the task of the human judges is actually simpler than the one being done by the classifier – the humans were asked only to determine the identities in a group, while the classifier must also examine a much larger amount of data to form those groups. Such a task is likely to be beyond the capabilities of human judges with the amount of sessions being considered here.

Overall, 45% of the clusters were determined by all judges to contain exactly one identity. If we relax the constraints and consider any cluster where at least one human judged it as being a single identity, we find that 55% of clusters qualify. However, quality is not consistent for all clusters; there were vast quality differences between clusters of different sizes. Small clusters (those containing fewer than 5 sessions) were far more likely to have only a single real identity than larger ones. Table 1 shows the relationship between cluster quality and cluster size. We believe the loose clustering used by our method is the reason for the poorer quality of the larger-sized clusters. Requiring only a single compatibility link allows for higher recall, though at the cost of also permitting a chain to form where S_i, S_j and S_j, S_k are compatible, but S_i, S_k are completely not.

The quality of the clusters produced by our method is shown in Figure 1. We would ideally want to see an average identity of 1, as this would mean the clusters did not include any incorrect matches. The results show that for clusters sizes up to 11, performance is fairly accurate, with no significant differences in number of distinct identities. As clusters grow beyond 11 sessions, they contain more identities; again, this is likely due to the loose clustering method. The clusters of size 20 include a system developer whose calls were correctly grouped together. Though this does validate our approach, in that it can find and cluster many calls from a single person, we must also note that system developers tend to be atypical users, which may explain the better quality at 20 sessions compared to the other large clusters.

4. Measuring Task Success for Different Identities

One interesting application of this work is that it allows us to examine how successful different identities are at using the dialog system. There is an automatic estimate of task success that is available from the Let's Go! system logs; this estimate is based on whether or not the system was able to provide bus schedule information to the user (as well as a few other minor factors). Though it is an overestimate, it has been shown to be highly correlated to true task success.

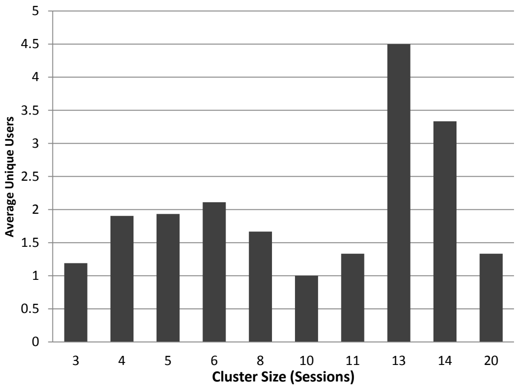

Figure 1: Average number of human-annotated unique identities within the automatically-formed clusters.

We expect that identities that use the system frequently would have a higher average success rate than infrequent or single users. This is due to user adaptation effects as well as familiarity with the dialog system capabilities; in essence, "veteran" users are implicitly trained how to use the system. To test whether this expectation is true, we examined the estimated success rate of all identity clusters, except for 3 which were determined to be system developers making test calls. Though these 3 clusters correctly group calls from a single user identity together, the sessions themselves did not represent normal usage of the dialog system and no meaningful information can be inferred from the success rate of such calls, which justifies their exclusion.

Figure 2 shows the average (estimated) success rate for clusters of at least a certain size; that is, size 10 shows the performance of all clusters with 10 or more sessions. It is clear that the trend is for increasing success rate as cluster size increases. Since a larger cluster size implies a more frequent (and more experienced) user, this confirms the expectation that such users are more effective at using the dialog system.

We also considered the effects of *consecutive callers* on success rate. As previously described, Let's Go! users will frequently hang up and call back immediately if they do not have a successful interaction. This can have a dramatic impact on success rates if several calls in a row from the same user are unsuccessful. To see what impact, if any, consecutive callers had on success rates, we excluded all sessions from a cluster that were sequential, using only the final session of a consecutive call set to determine success rate. This has the side effect of reducing cluster sizes as well. After excluding the consecutive sessions, we performed the same analysis as above; these results are shown in Figure 3. Excluding the consecutive callers' sessions generally raises success rates across the board, though it shows reduced success rates for some of the larger clusters. It is not immediately obvious what the cause for that is. However, we hypothesize that consecutive call sessions may be more likely to be erroneously estimated as successful compared to

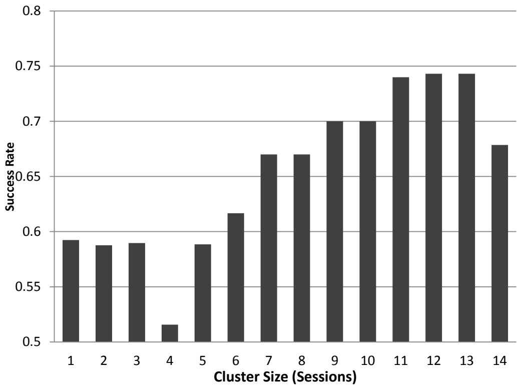

Figure 2: Average dialog task success rate for different cluster sizes. A size of 6 includes all clusters with 6 or more sessions.

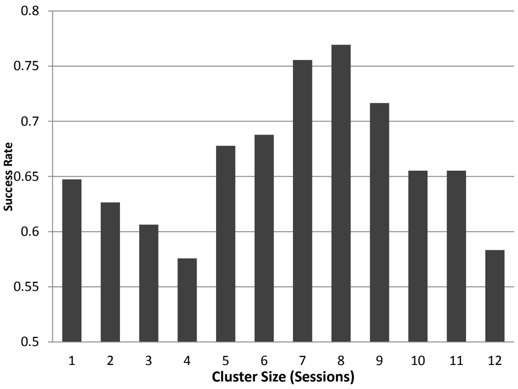

Figure 3: Average dialog task success rate for different cluster sizes, excluding non-final sessions from consecutive callers.

non-consecutive calls. We plan to more closely examine the distribution of incorrectly estimated successes to see if that explains the discrepancy.

In comparing cluster sizes against each other, large clusters (those with 10+ sessions) have a higher success rate than other cluster sizes, as well as non-clustered calls; this is statistically significant ($p < 0.03$). The task success rates for each cluster size are shown in Table 2.

5. Discussion and Future Work

Our method uses only acoustic data to create identity models. While such a minimalist approach allows for maximum flexibility, because this is a dialog application there are several other sources of information that are available to help identify users, including ASR output, dialog state information, and even what time of day the interaction occurs. Adding these sources to the acoustic information could help disambiguate similar users, as well as providing a broader model of identity to match against. For example, we have seen that there are users who often ask for the same bus at similar times across multiple days; models able to take such information into account will almost certainly outperform acoustic-only models.

Large clusters formed by our method are not as accurate, often containing more than one identity. As mentioned earlier, this is likely due to the loose clustering we used. Given our results, it may be preferable to require n compatibility links to add to an identity cluster, where n is proportional to cluster size, as opposed to either a single link or a larger but still constant number of links. However, the argument can also be made that for the purposes of dialog user identification, true user identity is not as important as *effective* user identity – that is, if two different people interact with a dialog system in a fundamentally similar way, treating them both as a single person may not be inappropriate. Further investigation of this is needed to determine whether this is true in practice.

Cluster Size	Success Rate
Unclustered	59.3%
3-9 Sessions	53.4%
10+ Sessions	**70.0%**

Table 2: Dialog task success rate for different cluster sizes. Bold indicates a statistically significant difference. Unclustered includes all sessions that do not belong to any identity cluster, and can be thought of as "single-use" callers.

Though our approach works well when analyzing a single month of dialogs, our method of matching clusters across months was not good at merging them into multi-month clusters. An effective inter-month matcher is vital for successfully identifying long-term users, which is arguably the most interesting pattern to recognize. Computational limitations drove the month-by-month combination approach we used in this work, which we believe is the reason we found very few multi-month clusters. We plan to examine other methods of dealing with larger amounts of session data, so that we can examine true long term users' experiences more effectively.

To make the most of this approach, we would want to implement a version that is capable of running online, in real time as a call comes in. This would enable a dialog system to do several different adaptations to the user, including ASR models and even dynamically changing the dialog task to be more suitable for the user. We are currently exploring how to make our approach efficient enough to run in real time without negatively impacting the user's dialog experience.

6. Conclusions

We have described a method for identifying unique spoken dialog users, using only acoustic information from their interactions. Our results show that our method is capable of identifying and clustering user identities from a month-sized set of dialogs, and can also reliably determine whether consecutive calls have come from the same identity.

7. References

[1] Q. Jin, "Robust speaker recognition," Ph.D. dissertation, Language Technologies Institute, Carnegie Mellon University, 2007.

[2] A. Raux, D. Bohus, B. Langner, A. Black, and M. Eskenazi, "Doing research on a deployed spoken dialogue system: One year of Let's Go! experience," in *Interspeech 2006*, Pittsburgh, PA, 2006.

[3] A. Black, P. Taylor, and R. Caley, "The Festival speech synthesis system," 1998, http://festvox.org/festival.