SPEAKER DIARIZATION: A PERSPECTIVE ON CHALLENGES AND OPPORTUNITIES FROM THEORY TO PRACTICE

Kenneth Church[1], Weizhong Zhu[1], Josef Vopicka[2]
Jason Pelecanos[1], Dimitrios Dimitriadis[1], Petr Fousek[2]

IBM, [1]USA and [2]Czech Republic

ABSTRACT

This paper discusses some challenges and opportunities in developing a speaker diarization system for operation on real world call center telephony data. We contrast some of the differences between a standard data set akin to NIST evaluations and those found in call centers. In exploring these differences we discovered vulnerabilities and proposed changes to address them.

In moving from theory into practice we introduce two tasks in which speaker diarization and recognition can be leveraged. First, we show that speaker diarization and recognition systems can be integrated to find the common speaker (the call center agent) across multiple calls and consequently their role. Furthermore, once the role is determined the corresponding speech recognition output can be analyzed to determine the type of support call.

Index Terms— Speaker Diarization, Speaker Recognition, Role Modeling, Call Center Data

1. INTRODUCTION

This paper discusses some of the challenges, as well as opportunities, that we encountered while transferring established diarization techniques into a real world call center application. We expected to encounter obvious challenges that don't tend to come up as much in standard data collections (e.g., Callhome [1]): music, tones (e.g., touch tones[1]), transfers to additional speakers, recorded voices, etc. But in addition to obvious unpleasant realities, we were pleasantly surprised by unexpected opportunities to collaborate with colleagues in other disciplines on new applications that go beyond standard diarization such as role modeling (who is the customer and who is the agent) and script classification (which calls are using which script).

To date, much of the diarization literature tends to focus on relatively clean data, but the real world isn't so clean. Our call center data exhibits strong statistical associations between speakers, roles, gender, telephone extensions and scripts. These observations create opportunities for two types of novel work: (1) applications of established diarization methods to new tasks such as role modeling and script modeling, as well as (2) combinations of diarization methods with methods borrowed from other fields such as text classification.

Much research has investigated how to improve speaker diarization on telephony (more recently [2, 3, 4, 5, 6]) and non-telephony data [7, 8, 9, 10] such as broadcast news, meeting-room and conferencing audio. There are also reviews covering these multiple domains [11, 12]. In particular, we are interested in the telephony related data sets and it is noted that these sets typically have well prescribed structures. For example, the NIST [13] and LDC [1] telephony data for diarization is mostly *2-speaker* conversational telephony data with a relatively smaller subset containing more speakers. There is also relatively little music. In this work we explore speaker diarization for call center data where company representatives are offering customers access to technical support. This task introduces new challenges such as on-hold music, pre-recorded speech, synthesized speech, tones, and multiple speakers. In this data such attributes represent the norm and are not the exception. With the goal of improving the analysis of this type of data we proceed with initial steps and some adjustments toward performing speaker diarization on this data. Building from this we share some additional analytics that can leverage the outputs from speaker diarization.

For call center data there are a number of issues which can affect diarization performance. There can be significant noise and the recording can contain on-hold background music. It is well known that, for optimal clustering, segments selected by the speaker diarization system should be representative of the speaker and not capture other artifacts [14]. These artifacts can cause a system to hallucinate additional speakers. The work by Anguera [15] demonstrated that building a better speech/non-speech detector provided significant benefits to speaker diarization performance. We note also that in the call center data we observed, there can be significant differences between the average intensity levels of the speakers on the call. This presents additional challenges for speaker diarization in the sense that is the low-level speech simply background cocktail party speech or is it a speaker the system needs to pay attention to. There are also challenges related to having responsive online[2] systems.

With the advancement of offline speaker diarization on telephony data and the evolution of conversational systems, there is also a growing interest for left-to-right (L2R) online systems. The work of [16] showed that by running the software on a GPU based system, with each new candidate segment, the system was able to recluster all segments up to the current point in time in realtime. In [17, 18], the authors proposed an initial *warmup* period after which the system learns to differentiate between the speakers later in the call. The approach is effective when the warmup period is representative of the rest of the call but unfortunately, it is common in call centers to transfer the call to some other agent well after the warmup period. Our recent work [6] evaluated a strict L2R system that classified each turn in a greedy fashion without the ability to go back and update previous decisions. This work built upon the work of Shum [2] by having an adaptive subspace to discriminate between speakers as the call progresses. This system was evaluated on Callhome English data [1] with promising results. However, when this same setup was evaluated on our call center data, we realized that the system was too sensitive to the initial conditions of the recordings. The greedy

[1] https://en.wikipedia.org/wiki/Dual-tone_multi-frequency_signaling

[2] https://en.wikipedia.org/wiki/Online_algorithm

heuristics usually work well, but failures are too common.

2. CALL CENTER DATA BACKGROUND

Call center calls are very different from standard NIST telephony speaker diarization evaluations [13]. Most of the calls in the call center scenario involve two parties, an agent (A) and a customer (C), though it is not unusual for customers to be placed on hold or transferred elsewhere. Transfers and holds can be particularly challenging for online systems because, unlike standard NIST evaluations, the beginning of the call might involve different speakers from the end of the call. Holds often involve tones and pre-recorded voices not found during the warmup period. The agent after a transfer is typically different from the agent during the warmup period.

Typically, support calls follow a script. Many calls start off with the agent receiving a statement from the customer describing the nature of the problem, and end with the agent providing a ticket number to the customer for future reference. It is often necessary to know who said what to make sense of these calls. For example, in order to identify escalations, we need to know when customers say certain keywords. But these calls should not be confused with other calls where agents mention the same keywords in innocuous ways, typically ending calls with a boilerplate explanation of what to do if not 100% satisfied.

Voice logging systems vary considerably. Sometimes the two sides of the conversation are recorded on different channels, but unfortunately, the two sides are often combined onto a single channel requiring diarization to undo the damage.

Various realities create challenges for basic energy-based SAD (speech activity detection) [19]. Gain controls may not be set optimally; it is not uncommon for one speaker to be much louder than another. Loudness depends on other factors, as well. Given that customers are trying to solve a pressing problem, speech can be understandably emotional. It is not uncommon for the "silence" (non-speech) on the louder channel to be louder than speech on the softer channel.

It is widely understood in the research community [15, 14] that diarization performance depends on SAD. ASR systems (e.g., [20]) typically use much more information than just energy to segment speech. We have decided to use ASR for SAD. Performance is roughly comparable, as shown in Table 1. In addition, ASR SAD provides the additional benefit of providing timestamps that are compatible between ASR and diarization. Henceforth, we will use the term, *ASR SAD*, to distinguish this SAD from the basic energy-based alternative.

In this section we discuss some key approaches for building a more robust system for call center data.

3. ROBUSTNESS IN AN ONLINE SYSTEM

For building out a speaker diarization system for call center data, in addition to using robust ASR, it may also be useful to have realtime speaker labels as the conversation progresses. In previous work [6] we described an offline method based on the work of Shum [2] and extended it for online operation. Online operation is desirable because it is useful in some applications to output diarization labels in a causal fashion, well before the end of the dialog. The online approximation utilizes i-vectors [21] (a compact representation/embedding of speech segments into a vector space with $D = 64$ dimensions,

Type of SAD	Callhome Data % FA/Miss	Call Center Data % FA/Miss
Basic	9.6/0.7	9.2/13.5
ASR	10.6/0.6	12.5/11.1

Table 1. One might expect better SAD (speech activity detection) with ASR (automatic speech recognition) than a basic energy-based SAD [19], but we find that performance is merely comparable. The table compares basic and ASR-based SAD on two test sets. Both test sets were filtered to exclude calls with more than two speakers (109 of 120 calls met this filter for Callhome, and 59 of 71 calls for Call Center Data). We use the NIST scoring tool where errors of omission (miss) and errors of commission (FA or false alarms) are weighted by time. Thus, FA reports the percentage of time that non-speech is incorrectly accepted as speech, and miss reports the percentage of time that speech is incorrectly recognized as non-speech.

Type of SAD	Clustering Approach	Callhome Data % Spk	Call Center Data % Spk
ASR	Offline	2.6	2.7
ASR	L2R Prefix	3.9	3.1
ASR	L2R Greedy	4.2	6.6

Table 2. Three clustering methods are sorted from best to worst by % Spk (the percentage of time that one speaker is confused for another). SAD errors (FA and Miss) are not reported here because the same ASR SAD is used for all three methods. The proposed prefix method is a compromise intended to capture much of the performance of the offline solution, but to do so in a L2R (left-to-right) fashion.

designed to capture speaker characteristics),[3] and adapts them further to a speaker informative subspace as the call progresses. For each new homogeneous segment the algorithm greedily assigns the segment to existing clusters if a distance threshold is met.

While the greedy assumption works reasonably well on standard data collections such as Callhome, we encountered robustness issues when we tried to apply that method to call center data, where there are a number of new challenges such as tones and pre-recorded voices. In particular, the greedy assumption runs into trouble because these calls start with a computer voice whispering into the agent's ear a short summary of what happened in the Interactive Voice Response (IVR) system before the call was transferred to the agent. This computer voice will not be heard from again and therefore, i-vectors of this voice are not representative of the rest of the call.

3.1. Prefix Alternative

To avoid possibly problematic greedy assumptions, we propose the prefix method, a non-deterministic compromise intended to approximate the offline solution, but to do so in a L2R (left-to-right) fashion. Table 2 shows performance is as intended; the prefix method is better than the greedy method, but not as good as the offline solution. The difference is larger for call center data than Callhome data.

The prefix method starts with ASR SAD which segments the speech into T turns. The output from diarization is a string of T speaker labels. In the special case of two speakers, the diarization output is a bit vector of T bits, where the i^{th} bit indicates which of

[3] http://mistral.univ-avignon.fr/mediawiki/index.php/I-Vectors

Hamming Dist	Prefix (Time →)
0	112222112221212212
0	1122221122212122121
0	11222211222121221212
1	212222112221212212122
0	2122221122212122121222
1	11222211222121221212221
1	212222112221212212122212
0	2122221122212122121222122
0	21222211222121221212221222

Table 3. An example of the prefix method. There is a row for each turn. The prefixes show the assignment of speaker labels by the offline diarization algorithm to turns that have been seen so far. Note that hamming distances tend to be small.

the two speakers uttered the i^{th} turn. In the more general case, with k speakers, each of the bits is replaced with a categorical variable[4] with k possible values.

The prefix method avoids greedy heuristics as much as possible. After each turn, the prefix method applies the offline clustering solution to the audio turns that have been seen so far, as illustrated in Table 3. That is, after each turn, the prefix is a bit vector with the best assignment of diarization labels that we could come up with using our best method (the offline approach). When it becomes necessary to output speaker labels, we use the prefix to do so. It may be required to output labels after every turn, every few turns, or before a set time has elapsed.

To avoid undesirable flip-flopping, we introduce a *reconciliation* process. Because there is no special meaning of 1 and 2 in the offline output, and all the bits in the offline solution can be flipped without changing the error rate, there is a risk that the meaning of the labels could flip-flop over time. In the more general case with more than two speakers, instead of just two possibilities (flip all the bits or not), we need to consider permutations of speaker labels. For each permutation we calculate the hamming distance to the prefix for the previous turn. Reconciliation uses the permutation that is most consistent with the previous turn, as illustrated in Table 3.

The prefixes in Table 3 are remarkably consistent with one another, as indicated by the small hamming distances. Even so, note that the first column is somewhat unstable, which has more 1s at the top and more 2s at the bottom. In this case, the instability is caused by a computer voice and this was causing assignments to be mostly nonsensical (not shown) for the greedy left-to-right online method [6]. The beginning of these calls is not representative of the end, and therefore, i-vector centroids based on the beginning will not generalize well to the rest of the call. Table 3 shows that most columns are stable, but a few are very likely to flip back and forth. There are many reasons for flip-flopping in addition to the computer voice mentioned previously. Figure 1 looks at the number of turns processed so far (the length of the prefix), as well as the duration of the current segment. In general, stability improves with longer prefixes and longer turns.

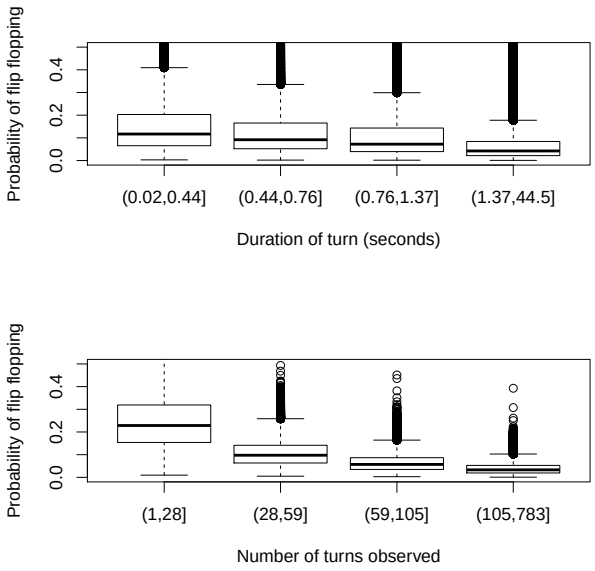

Fig. 1. Flip flopping (instability of diarization labeling across prefixes) depends on many factors including the duration of each turn (top) and the number of turns (bottom). Both plots split the data into quartiles. The probability of flip-flopping is highest for the first quartile and lowest for the last quartile. The top plot shows that flip-flopping is more likely for short turns (under 0.44 seconds) than long turns (over 1.37 seconds). The bottom plot shows that flip-flopping is more likely at the beginning (first 28 turns) than at the end (after 105 turns). The bottom plot shows that it is risky to depend too much on an initial warmup period. The odds are too high that the beginning of the call is not representative of the end of the call.

4. LEVERAGING DIARIZATION IN OTHER TASKS

4.1. Role Modeling

The diarization methods mentioned above can be used to address a range of other tasks such as role modeling. Diarization will distinguish one speaker from another (e.g., S1 vs. S2), but in the call center scenario, we want to know which of these two speakers is the customer (C) and which is the agent (A). At some point in the future, we would like to consider the more general case where diarization can output more than two speakers and more than two roles. In that case, the role modeling task is to find a mapping between diarization outputs and dialog roles. In the special case of just two speakers and just two roles, we need just a single bit to determine this mapping, though obviously, the more general case is harder than the special case discussed here.

We have two solutions to the special case, one based on i-vectors and diarization-like methods, and another based on text classification and machine learning. The special case is actually remarkably easy; both methods work so well that it is difficult to measure performance. When error rates are small (less than a few percent), we need huge test sets (thousands of audio cuts) to see small differences between competing approaches. Unfortunately, since we don't have a large test set, we have had to introduce various workarounds to estimate performance.

The i-vector method starts with a training set of $k = 10$ calls where there is a single agent that speaks on all k calls, and there are k different customers that speak on each of the k calls. Two-speaker diarization is applied to each of the k calls. I-vectors are trained on all clusters to produce a total of 20 i-vectors. Using agglomerative clustering (similar to [6]), the 10 closest clusters are found after a

[4] https://en.wikipedia.org/wiki/Enumerated_type

constraint is considered. The constraint is that only one i-vector from each call can be assigned to the 10 clusters. A final i-vector is then calculated from the data relating to the 10 clusters. This i-vector representation is used to directly detect (using speaker recognition) which speaker is the agent in the agent's test set of diarized calls. This process can be generalized to work in a more practical setting, where we start with a corpus of calls from a call center, and find agents by looking for speakers that speak on many calls (though far from all calls).

The text classification method starts with the results of the i-vector method to train a text classifier. The classifier uses both word unigrams and bigrams. The task is to learn a single bit for each call. Given labels from the speaker diarization system, we have a bag of words for the agent and a bag of words for the customers. These two bags are input features to Libshorttext [22], a popular supervised machine learning package. The classifier learns which keywords are associated with agents and which are associated with customers.

To test the text classification system we used a 1,352 call set which was split into a 1,216 call training part and a 136 call validation part. The recordings from the output of the diarization system were assigned a role by the diarization process described previously. The best result using text classification produced a classification error rate of 1.1%. Consequently, there were 3 calls associated with an incorrectly assigned role; 2 of the 3 errors were attributed to a single call, where the diarization process failed and assigned roles incorrectly. The remaining error was caused by the call content which may be characterized as an out of domain problem the help desk was dealing with. In review, speaker diarization can be used to provide training data for a text classification system that can learn speaker roles. The system can later classify the roles based only on a single call independent of the agent.

This method should work well because it is common for agents to use different words from customers. Agents are typically working from a well-rehearsed script unlike customers who are typically answering these questions for the first time. It is noted that ASR word error rates tend to be higher for customers than agents, as well.

Simple keywords are surprisingly powerful for role modeling. That is, the agent is much more likely to use deferential words like "please", "sorry", and "sir". In our data set, we observe that customers are likely to be male and agents are likely to be female. Agents are more likely to use certain technical words that customers are unlikely to know. Additionally, agents use words like "hold" and "transfer" more than customers.

One of the strongest signals is the name of the agent. Most agents in our corpus introduce themselves at the beginning of the call: "Hi my name is Ashley... will this be for a new or existing request". Some customers (about 10%) will respond by addressing the agent by her name, "Hi Ashley", before answering Ashley's question, but most customers answer the question without saying "Ashley". Thus, the name of the agent is an excellent feature for role modeling, at least in the case of helping us learn about Ashley's calls. This name feature can also be used to estimate error rates for our i-vector system as illustrated in Table 4.

As mentioned above, both i-vector and text classification methods are so effective for role modeling that it is difficult to measure error rates. Given small error rates of a couple percent, we couldn't afford to build a large enough test set to measure error rates in the standard way. Admittedly, a concern with Table 4 is that there is just one agent in that test. To check if that is a problem, we looked at the phrase, "are you calling," over 3 extensions with at least 1000 calls per extension. Assuming this phrase should be assigned the A role (since it is part of the agent's opening prompt), the error rates vary by extension from 0.15% to 2.5%. Role modeling is often made easier by gender statistics. Typically, the customer is male and the agent is not. The high error rate of 2.5% is associated with an exception where the agent happens to be male.

# Calls	Assignment	ASR Output
1243	Agent ✓	**h**i my name is Ashley will this be for a new or an existing request
9	Customer ✗	**h**i my name is Ashley will this be for a new or an existing request
9	Agent ✓	**m**y name is Ashley will this be for a new or an existing request
8	Customer ✗	**h**i my name is Ashley will this be for a new or an existing request
7	Agent ✓	**h**i my name is Ashley

Table 4. The fact that agents typically introduce themselves on the first turn can be used to estimate role modeling performance. These top-5 opening turns should obviously be labeled as *Agent*, but 1.3% of these calls are incorrectly labeled as *Customer*.

Skill ID	A	B	C	D	E	F
A	75	0	0	0	0	0
B	0	131	0	1	0	0
C	0	0	291	1	0	0
D	0	0	0	1565	0	8
E	0	0	0	0	77	5
F	0	0	0	40	20	4959
Accuracy	100%	100%	100%	97%	79%	100%
# Calls	75	131	291	1607	97	4972

Table 5. Confusion matrix for skill (types of support) classification.

4.2. Script Modeling

The text classification method mentioned earlier can be used for a variety of tasks beyond role modeling. In our help desk data, there are 6 types of support (skills/scripts). We have assigned skills to 71,327 calls based on some manual work and heuristics. We split the data into a partitioning of 10% validation data and 90% test data covering all phone line extensions. Extensions are usually used by one agent giving support for one category. The text classification (utilizing the transcribed and diarized output) also works very well on this task. The average per-skill accuracy is 96% while the overall per-call accuracy is 99%. Confusion matrix results are shown in Table 5.

5. CONCLUSIONS

This paper discussed some challenges and opportunities in transferring speaker diarization to a real world call center. We found some robustness issues with some techniques that worked well on standard datasets. We elected to apply diarization after ASR, and use ASR for SAD (as opposed to energy-based SAD). We proposed the prefix method to capture much of the performance of the offline solution, but to do so in a L2R (left-to-right) fashion. The prefix method is particularly beneficial in the call center case, where large improvements over the greedy method of 3.5% absolute (6.6% $\rightarrow$ 3.1%) were reported in Table 2. Furthermore, the paper established a link between speaker diarization and new tasks such as role modeling and script classification, with opportunities for interdisciplinary collaboration.

6. REFERENCES

[1] Linguistic Data Consortium, "LDC speech group website," http://www.ldc.upenn.edu, 2016.

[2] S. Shum, N. Dehak, R. Dehak, and J. Glass, "Unsupervised methods for speaker diarization: An integrated and iterative approach," *IEEE Transactions on Audio, Speech, and Language Processing*, 2013.

[3] G. Sell and D. Garcia-Romero, "Speaker diarization with PLDA i-vector scoring and unsupervised calibration," in *IEEE Spoken Language Technology Workshop*, 2014.

[4] M. Senoussaoui, P. Kenny, T. Stafylakis, and P. Dumouchel, "A study of the cosine distance-based mean shift for telephone speech diarization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 1, pp. 217–27, 2014.

[5] D. Reynolds, P. Kenny, and F. Castaldo, "A study of new approaches to speaker diarization," in *Interspeech*, 2009, pp. 1047–1050.

[6] W. Zhu and J. Pelecanos, "Online speaker diarization using adapted i-vector transforms," in *IEEE ICASSP*, 2016, pp. 5045–5049.

[7] M. Zelenák, H. Schulz, and J. Hernando, "Speaker diarization of broadcast news in Albayzin 2010 evaluation campaign," *EURASIP Journal on Audio, Speech, and Music Processing*, 2012.

[8] J. Luque, X. Anguera, A. Temko, and J. Hernando, *Multimodal Technologies for Perception of Humans*, vol. 4625, chapter Speaker Diarization for Conference Room: The UPC RT07s Evaluation System, pp. 543–553, Springer Berlin Heidelberg, 2008.

[9] C. Barras, X. Zhu, S. Meignier, and J. Gauvain, "Multistage speaker diarization of broadcast news," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, pp. 1505–1512, 2006.

[10] N. Hieu, *Speaker Diarization in Meetings Domain*, Ph.D. thesis, School of Computer Engineering of the Nanyang Technological University, 2015.

[11] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Transactions on Acoustics, Speech, and Language Processing*, vol. 20, pp. 356–370, 2012.

[12] S. Trantor and D. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Transactions on Audio, Speech, and Language Processing*, pp. 1557–1565, 2006.

[13] National Institute of Standards and Technology, "NIST speaker recognition evaluation website," *http://www.itl.nist.gov/iad/mig/tests/spk/*, 2016.

[14] D. Reynolds and P. Torres-Carrasquillo, "Approaches and applications of audio diarization," *ICASSP*, pp. 953–956, 2005.

[15] X. Anguera, M. Aguilo, C. Wooters, C. Nadeu, and J. Hernando, "Hybrid speech/non-speech detector applied to speaker diarization of meetings," *IEEE Odyssey - The Speaker and Language Workshop*, 2006.

[16] G. Friedland, "Using a GPU, online diarization = offline diarization, Report TR-12-004," Tech. Rep., ICSI, 2012.

[17] O. Toledo-Ronen and H. Aronowitz, "Confidence for speaker diarization using PCA spectral ratio," *Interspeech*, 2012.

[18] H. Aronowitz, Y. Solewicz, and O. Toledo-Ronen, "Online two-speaker diarization," *A speaker odyssey*, 2012.

[19] S. Sadjadi and J. Hansen, "Unsupervised speech activity detection using voicing measures and perceptual spectral flux," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 197–200, 2013.

[20] G. Saon, H. Kuo, S. Rennie, and M. Picheny, "IBM 2015 English conversational telephone speech recognition system," *Interspeech*, pp. 3140–3144, 2015.

[21] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[22] H Yu, C Ho, Y Juan, and C Lin, "Libshorttext: A library for short-text classification and analysis," *Rapport interne, Department of Computer Science, National Taiwan University. Software available at http://www. csie. ntu. edu. tw/˜ cjlin/libshorttext*, 2013.