2004 IEEE 11th Digital Signal Processing Workshop
& IEEE Signal Processing Education Workshop

UNICA - A UNIFIED CLASSIFICATION ALGORITHM FOR CALL PROGRESS TONES

Manish Marwah and Sharmistha Das

Avaya Labs
1300 West 120th Ave., Westminster, CO 80234
and
University of Colorado, Boulder, CO 80309

ABSTRACT

To increase productivity in an outbound call center, a predictive dialer system is used to automate dialing and to route calls to available agents. Such a system needs to monitor the progress of a call so that the call is sent to an agent depending on how it completes, for example, it may be desired that calls be routed to an agent only if they are answered by a live person. A call classifier, which is part of such a system, is involved in detecting how a call completes. Typically a call classifier contains various DSP algorithms including ones for classifying different kinds of call progress tones (CPTs) such as busy tone, reorder tone etc.

In this paper, we propose a generic algorithm, called UniCA (Unified Classification Algorithm) that can be used for classifying any CPT. It is particularly useful for distinguishing between CPTs that have the same frequency components or in situations where the frequency information is not available to the CPT classifier. In addition, the algorithm is efficient, deals intelligently with ambiguities and noise, and can correctly distinguish between any two theoretically distinguishable CPTs. A version of this algorithm has been implemented and is part of a commercial product. Our implementation was tested with a variety of real and simulated CPTs. The results show that our algorithm performs better than earlier algorithms [1].

1. INTRODUCTION

Call progress tones (CPTs) like busy tone, reorder tone, ringback tone etc. are used for in-band signaling in telecommunications. They convey important call progress information to the user. They are also vital for a predictive dialer system [2] to determine how a call is answered. A predictive dialer system is typically used in outbound call centers to automate placement of outbound calls, detection of how the calls complete and intelligent routing of calls to agents with the goal of minimizing both the number of abandoned calls (valid calls which could not be delivered to an agent) and the number of idle agents at any time. Such a system provides various tunable parameters. For example, the system can be configured such that an agent is assigned to a call only after it is determined that the call has been answered by a live person. If the call ends in a busy or reorder tone, or, it is answered by an answering machine, it can be automatically discarded (or, in case of an answering machine, the system can leave an appropriate message). The classification of how a call completes is made by a call classifier which is an essential part of predictive dialer systems.

An important part of the call classifier is a CPT classifier (see Section 2 for details). Implementing an efficient and accurate CPT classifier can be tricky, especially for tones[1] which have different cadences but the same frequency components. CPTs are usually country specific and each country has a number of different types of CPTs. It is inefficient to have separate CPT classifiers for CPTs of different types or for CPTs from different countries. Moreover, the classifier needs to be generic enough so that only minimal configuration changes are required if the definition of a CPT changes.

In this paper, we propose a generic, unified CPT classification algorithm, which can be used to classify any CPT, irrespective of the CPT type or country. Furthermore, the proposed algorithm is robust and efficient. It is robust since it deals intelligently with ambiguities and noise; and efficient, since it classifies the incoming CPT, in almost all practical situations, in one cycle of the tone, where one cycle is the duration of the basic unit of the tone.

The following section provides a short description of a traditional call classifier. In Section 3, we discuss, in detail, the proposed algorithm, UniCA (Unified Classification Algorithm) for CPTs. This is followed by a section which summarizes the advantages of UniCA. Some implementation details are provided in Section 5. And, finally the conclusions are presented in Section 6.

---

[1] Any occurrence of the term tone refers to a CPT in this paper.

2. CALL CLASSIFIER

A traditional call classifier can be considered to have three essential components — a voice detector, a voice discriminator and a CPT classifier. The voice detector detects if any voice is present in the incoming signal, the voice discriminator distinguishes between a live voice and a recorded answering machine voice, and, the CPT classifier classifies tones. These components of a call classifier are shown in Fig. 1.

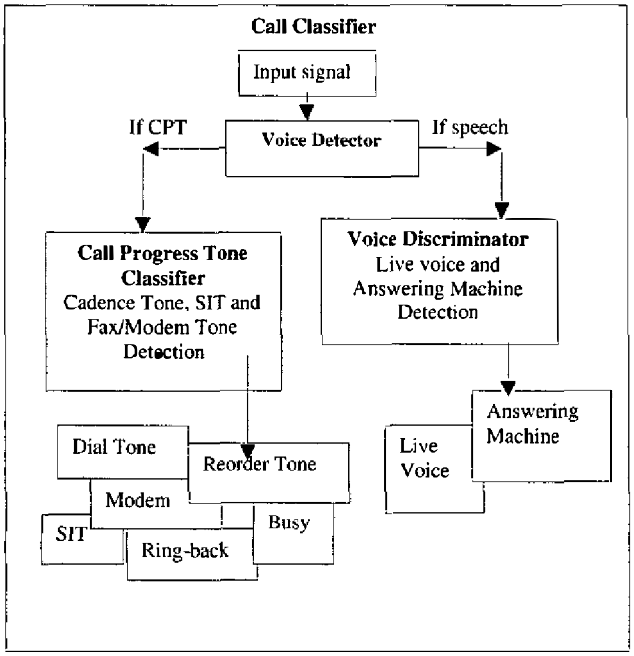

Fig. 1. Components of a call classifier.

The input signal first passes through the voice detector. If it determines that it is a speech signal, it is directed to the voice discriminator, otherwise, it is assumed to be a tone and sent to the CPT classifier.

2.1. Call Progress Tone Classifier

Call progress tones can be categorized as:

1. **Cadence Tone.** A repeating, periodic CPT like busy, reorder, or, ringback tone. The basic repeating unit (also referred to as one cycle) of such a tone usually consists of alternate segments of energy and silence. A special case of a cadence tone is continuous cadence tone, which does not have any intermittent silence periods, keg. dial tone in the U.S.
2. **Special Information Tone (SIT).** A short, non-repeating CPT made up of three contiguous energy segments. It usually precedes a recorded announcement about an exception condition e.g. a nonexistent phone number.
3. **Modem/Fax Tone.** A CPT generated by a modem or a fax machine.

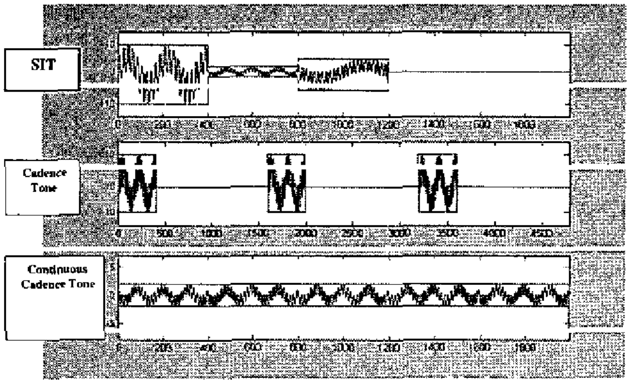

Fig. 2. Examples of some CPTs.

Some CPTs are shown in Fig. 2. CPTs may vary in both frequency components and cadence for different countries. For example, ringback tone in the U.S. is a cadence tone consisting of 1 sec of energy (a dual frequency of 440 and 480 Hz) and 3 sec of silence, whereas in Japan it consists of 0.5 sec of energy (400Hz modulated by 20 Hz) followed by 1 sec of silence. An interesting example is Iraq, where the ringback tone is a continuous cadence tone of a single frequency (400 Hz).

A call classifier may use a different mechanism for classifying each type of CPT. For example, in [3], one algorithm is used for detecting international cadence tones, and a different hard-coded, state-machine based algorithm is used for U.S. domestic cadence tones; SIT tones are detected by auto-correlation analysis in a DSP; while fax and modem frequencies are detected using DSP matched filters. For further details on call classifiers, see [4].

3. UNIFIED CLASSIFICATION ALGORITHM FOR CPT (UNICA)

Our proposed algorithm – UniCA – can be used to identify any generic CPT. Tones are specified as a series of segments. A segment is a continuous duration of energy with same frequency components (also referred to as an energy/pulse segment or a P-segment) or a continuous duration of silence (also referred to as a silence segment or a S-segment). The length of a CPT is defined as the number of segments in one cycle of the tone. For example, the U.S. domestic busy tone has two segments, a P-segment (with frequency components 480Hz and 620Hz) lasting 500ms and a S-segment also lasting 500ms. Thus, its length is two. UniCA uses a database of CPTs (tone table) against which it compares any incoming CPT and attempts to classify it as one of tones in the database. If it is unable to classify the signal, it declares it as non-classifiable. The actual algorithm is presented in the following section. Next, we discuss some of the issues involved, namely, how many segments need to be matched to unambiguously identify a CPT, how are glitches and noise handled, and finally, how are ambiguous matches dealt with.

3.1. Description of UniCA

The algorithm to classify CPTs is a parallel search algorithm with no backtracking. It assumes that the incoming signal could start at any of the segments of any of the tones present in its tone table (database). It uses a data structure, which we will refer to as a search engine, to keep track of the state of the parallel search. An engine points to a segment in a tone and remembers how many segments it has already matched. If it matches the segment it points to, it moves to the next segment in the tone, else, it drops out and is not used anymore. The engines that continue matching are referred to as valid and the ones that get eliminated are called invalid. Initially, all the engines are valid. The total number of engines is equal to the sum of the lengths (number of segments) of all the tones in the tone table. The number of CPTs in the tone table are not typically be very large (less than around 20) and for each tone usually there are at most four segments. Further, an engine takes very minimal memory resources. This makes the engines quite lightweight in terms of resource utilization.

A slightly simplified pseudo-code of the algorithm is presented below:

1. Create a search engine for each segment of each tone present in the tone table. Mark them as valid.
2. Wait for an incoming segment (energy or silence) to be over.
3. Try to match this segment with all the segments that valid search engines point to. The match involves both a time interval and a frequency component match (if the frequency information is available).
4. Mark all the engines that do not match as invalid.
5. If no valid engines remain, declare the incoming signal as non-classifiable and the algorithm terminates.
6. For all the engines that are still valid, compute a goodness of fit value and add it to a running sum for that engine.
7. If sufficient number of segments have been matched, pick the engine that has the best goodness of fit. At this point we classify the incoming tone as the one to which that engine belongs and we are done.
8. If sufficient number of segments have not been matched, then move all the valid engines to the next segment in the tone.
9. Go to Step 2.

A few notes on the pseudo-code:

1. The wait in Step 2 also involves handling glitches and noise.
2. The goodness of fit value, in Step 6, is the square of the Euclidean distance between the two segments (namely, the one that the engine points to and the incoming one).
3. The sufficient number of matches in Step 7 is computed theoretically based on the number of segments in all the tones that have valid engines associated with them and is discussed in detail in Section 3.2.

A state diagram for UniCA is shown in Fig. 3

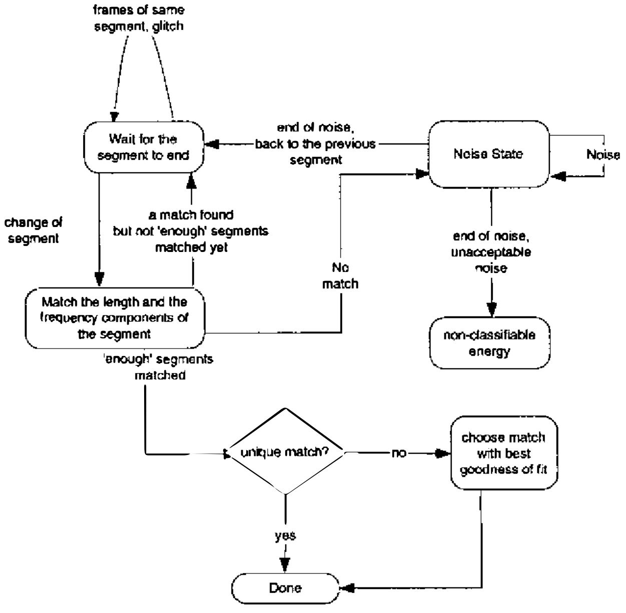

Fig. 3. A state diagram for UniCA

Now, we run through an example that illustrates how the algorithm classifies a CPT. Assume that there are two CPTs in the tone table (CPT1 and CPT2). These two tones and the incoming tone which needs to be classified are shown in Fig. 4. The energy segments (P) and silence segments (S) have also been labeled in the figure.

The following summarizes the state of the algorithm after each segment of the incoming CPT is received:

Initial State: There are six engines, each pointing to a segment of a CPT in the tone table.

After segment P_{i1} is received: All engines pointing to silence segments are eliminated. Further, the engine pointing to P_{21} also becomes invalid. Only engines pointing to P_{11} (lets call it E_1) and P_{22} (lets call it E_2) match and are incremented to point to the next segment, that is, now E_1 points to S_{11} and E_2 points to S_{22}.

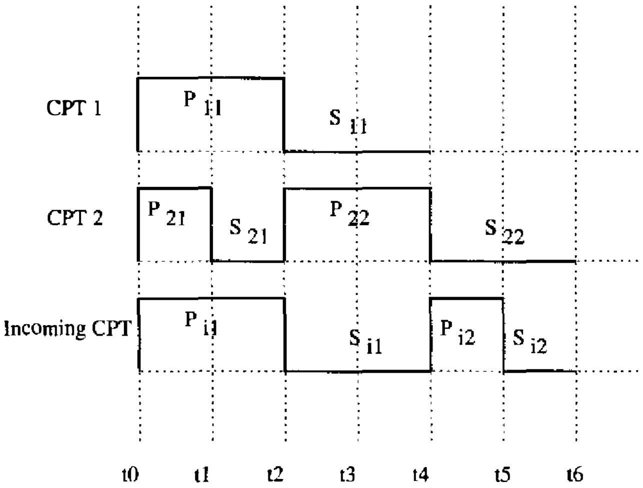

Fig. 4. Tones CPT1 and CPT2 are in the tone table (tone database). Incoming CPT is the tone that is needs to be classified.

After segment S_{i1} is received: Both engines E_1 and E_2 match and hence remain valid. They now point to segments P_{11} and P_{21} respectively.
After segment P_{i2} is received: E_1 fails to match and is eliminated. E_2 continues to match and is the only remaining engine. It now points to S_{21}.
After segment S_{i2} is received: E_2 matches again and since a sufficient number of segments have been matched, the incoming signal is declared to be CPT2.

3.2. Sufficient Number of Matches Required

In this section, we explore the question – What is the sufficient number of segments that need to be matched before we can declare that we have matched a tone? We are only concerned with repeating tones here, and, as mentioned before, the incoming signal can start at any segment in the tone. Now, if the maximum length of a tone in the tone classifier database is n, it may seem, on a cursory look, that at most n matches are required to uniquely classify an incoming tone. However, this is not true as shown by the following example.

Here we use letters A and B to denote two different segments. Consider that the tone database consists of two tones of lengths 3 and 4 as shown below.

$$BAB \tag{1}$$

$$ABBA \tag{2}$$

Further, assume that the incoming signal is ABBABBA... As we consider each segment in turn, we find that an engine in both tones matches all the first four incoming segments, namely, ABBA. These are the second engine, the one starting at A, in tone (1) and the first one in tone (2). Thus, we find that in this case four matches are not sufficient. If we consider one more segment, we find that the incoming tone is actually tone (1). The fifth incoming segment is B, which matches the engine in tone (1). The engine in tone (2) does not match since it points to A, thus dropping out and we have a unique match in tone (1). Thus, five segments were required here to unambiguously identify the incoming tone.

The above discussion leads to the following question: In general, given two tones T_1 and T_2 of lengths l_1 and l_2 respectively, what is the sufficient number of segments that need to be matched to uniquely identify an incoming tone as one of the two?

Without loss of generality, assume $l_1 > l_2$. Also, the fact that the incoming tone can start at any segment is inconsequential here, since for each tone we can simply add more tones in the database such that we have tones starting at each of its segments. Thus, we can assume that we only have to start the match from the beginning of a tone. Consider instances of T_1 and T_2, with both the tones extending forever, repeating their basic units. Now, the sufficient number of segments that need to be matched is the length, starting from the beginning of each tone, till the point where the segments of both the tones differ. If l_2 divides l_1, then clearly this value is l_1, since in this case if the segments continue to match till l_1, then the two tones are identical. In the case when l_2 does not divide l_1, we use a divide and conquer algorithm. (We do not provide a rigorously proof of the algorithm here). This algorithm is shown graphically in Fig. 5. We reduce the problem for tones of lengths l_1 and l_2 to that for tones of length l_2 and (l_1 mod l_2). The corresponding recurrence relationship, $m(l_1, l_2)$, for the sufficient number of matches required is given below.

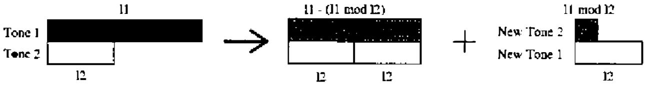

Fig. 5. A pictorial depiction of the recurrence relation for computing the sufficient number of matches.

$$m(l_1, l_2) = \begin{cases} m(l_2, l_1) & \text{if } l_2 > l_1 \\ l_1 & \text{if } 1_1 \bmod l_2 = 0 \\ l_1 - (l_1 \bmod l_2) & \\ + \, m(l_2, l_1 \bmod l_2) & \text{otherwise} \end{cases} \tag{3}$$

Using this to compute the maximum number of matches required for the example we considered earlier (for two tones with lengths 3 and 4), we get $m(4,3)$ equal to 6. So although we were able to distinguish between the two tones using 5 matches that is not sufficient. There could be another example of tones of lengths 3 and 4 where 5 matches are not enough as shown below.

$$BAAA \tag{4}$$

$$AAB \tag{5}$$

Consider the incoming signal to be AABAAAB... As can be seen it requires 6 matches to classify this as tone (4). However, since 6 is the sufficient number of matches required, it is not possible to construct an example of two tones with lengths 3 and 4 where more than 6 matches are needed.

Although we have only considered two tones at a time here, it is quite straightforward to conceptually extend it to more number of tones by simply computing the sufficient value for all the pairs of tones and then choosing the highest value among them. This would give us the overall maximum for the entire set of tones.

3.3. Ambiguous Matches

One may wonder why, in the previous section, we spent so much effort to determine the maximum number of matches required to tell apart two cadence tones, because, one strategy could be to continue matching segments as long as valid engines exist for more than one tone. And, the moment we have valid engines for only one tone, we stop and declare that we have a match.

The above would have been reasonable if we always had pure signals with no variation in the tone signals. However, that is not true in real life, and, to account for the variations in the time duration of a segment in a tone, a range is used instead of just specifying a single value. If these ranges overlap for two tones, it is possible that the incoming signal continues to match multiple number of tones. In such a case, we need to address two issues — (1) when to stop matching, and, (2) when we do stop, how to uniquely and intelligently resolve the ambiguity. For (1), we use the result from the previous section and stop when we have matched sufficient number of segments. For (2), we resolve the ambiguity by choosing the tone with the best goodness of fit value.

4. ADVANTAGES OF UNICA

In this section we summarize the advantages of UniCA.

Efficient: UniCA never takes more than the sufficient number of matches required for the tones in the database. In most cases, it is able to identify the incoming CPT much sooner, typically taking only one cycle of a CPT irrespective of where in the tone the incoming signal starts at.

Generic Unified Approach: UniCA can be used to classify any theoretically distinguishable tones. Some situations where UniCA will work successfully while other algorithms may fail are — (1) cadence tones, where one tone is a subset[2] of another tone (2) the final part of one cadence is identical to the initial part of another cadence. (Here, it is assumed that the tones either have identical frequencies, or, the frequency information is not available.)

Performs Well in the Presence of Noise: The segments of a cadence can be defined as a range instead of a single value to account for noise. UniCA uses the goodness of fit criterion to resolve any ambiguities in such cases. It also removes glitches from the incoming signal.

5. IMPLEMENTATION OF UNICA

A version of UniCA has been implemented. This includes the search engine framework, recognizing cadences, computing the goodness of fit value, dealing with ambiguities, and, handling glitches and noise. This has been implemented in C++ in a VxWorks environment. We have tested extensively with various real and made-up CPTs. We found a number of test cases where UniCA correctly classified noisy tones whereas an earlier algorithm did not. For more details, see [5].

6. CONCLUSIONS

In this paper, we presented a new CPT classification algorithm (UniCA), which has significant advantages. The new algorithm can be used for any generic tone, not just a particular type of tone. Its primary advantage lies in distinguishing between tones with identical frequencies or if the frequency information is missing. The algorithm is efficient, flexible and handles noisy signals well by intelligently resolving ambiguities. An implementation of UniCA has been incorporated in a commercial product and performs better than the earlier algorithms used.

7. REFERENCES

[1] L. J. Roybal, "Signal-Recognition Arrangement Using Cadence Tables," United States Patent, 5,719,932.

[2] "Avaya Predictive Dialer System," www1.avaya.com/enterprise/whitepapers/gcc0938.pdf

[3] L. J. Roybal, "Function Specification for TN744D Global Call Classifier," Internal Avaya Document, 1995.

[4] "DEFINITY ECS Call Classification White Paper," Avaya White Paper (to be publically available soon).

[5] M. Marwah and S. Das, "Generic Tone Classification Algorithm," Avaya Software Symposium, Nov. 2001.

---

[2] CPT T_1 is a subset of another CPT T_2, if T_1 is fully contained in T_2. For example, tone AB is a subset of tone AAB.