

Moving Beyond “I’m sorry, I didn’t get that.”

by
Jenni McKenzie
Paul Greiner
Sunil Issar

Table of Contents

Introduction..... 2

What’s an error?..... 2

Strategies..... 3

Utilize the grammars..... 5

Conclusion..... 6

Introduction

Writing the *positive* call flow is the truly fun part of voice user interface (VUI) design. However, miscommunication between the caller and the system can occur for a number of reasons. The caller may not have the information needed or may not know the system vocabulary. The mistake may be caused by technology limitations like the system not being able to differentiate between the caller's response and background talk. Without careful attention to the error handling strategy and prompts, callers can enter the death spiral of IVR hell to the point of hanging up out of frustration, requesting an agent, or having the system initiate the transfer to an agent. None of these outcomes is good for either customer satisfaction, brand image, or containment rates.

The goal of error strategies is to guide the caller to provide necessary information in a way that has a good chance of recognition with a high confidence score. For example, we can ask the caller to provide a more verbose response like "Dulles, Washington" instead of "Dulles." In this article, we will focus on two basic error categories: when the caller doesn't respond (no input), and when the caller does respond but the system is not able to understand the caller with a high degree of confidence (no match). Other errors resulting from call flow issues like an account and PIN not matching or a host transaction error will not be addressed. However, the basic methodology of understanding the possible cause and trying to address it still holds in many of these other situations.

What's an error?

Designing an error strategy always begins with the writing of well-crafted initial prompts that reduce errors in the first place. The issues surrounding the writing of error handling prompts reveal circumstances that vary wildly. There is no generic recipe for writing error prompts; no hard and fast rules that apply universally. The first step in writing effective error prompts is to understand why and how dialog "errors" occur.

Let's first consider the term "error." The word implies a departure from a smoothly flowing dialog having a single question-and-answer "turn" for each step through the callflow. In this sense, "error-free" dialogs are a lofty and unnecessary goal, given that they don't even exist in human-to-human communication. Asking your conversation partner to repeat or explain him or herself is not a detour from a human conversation, but rather a natural, integral part of it.

"That darn Laura makes me so mad!"

"Lauren?"

"No, Laura."

"Why's that?"



Yet the term “error” remains a convenient way to describe the two traditional dialog event categories of no input and no match. And we'll continue to use the term here.

First let's look at no input. It appears to be pretty simple on the surface; the caller doesn't say anything. But why? Are they not paying attention? Do they not have the information requested (like an account number)? Do they not understand what's being asked? Are the choices too similar and they're not sure which one is correct? Are they hoping that playing possum will get them to an agent? And sometimes (although rarely), the caller is talking but not loud enough. Each one of these situations would lead the designer to write a different type of error prompt. For example, if the caller did not respond to a menu selection, the error prompt should most likely provide some explanation and also additional choices if none of the options meets the caller's needs.

When we look at no matches, the breadth of possibilities grows even more. The first distinction is between callers trying unsuccessfully to use the system and those who aren't even trying. Callers may simply be restarting or self-correcting, attempting to give an in-grammar response. Out-of-grammar responses fall into a lot of categories as well. Callers may be rambling or supplying unanticipated pre-filler or post-filler. They might be trying to make the system error out and transfer them. Maybe side speech or background noise is causing the no match. Then there's the case of when the caller says something that's completely covered by the grammar, but just not recognized. Once again, each situation would lead the designer to a different reprompt.

So there are all these different types of errors, but we (designers) tend to have one way of trying to deal with them. The more we know about the error (the type, the circumstances, the user, the state you're in, etc.), the more we can tailor our handling of it. We have to look at guiding principles, remembering always that the more we know about the error, the more effectively we can craft specific verbiage to handle it gracefully.

Strategies

In the “old days” of VUI, we said things like, “I'm sorry, I didn't get that.” Do we really need to apologize? Do we even *need* a transition phrase? Sometimes changing the prosody can express equally well what robotic transition phrases have done in the past. For example asking, “Whaaat was that account number?” with a drawn out “what” does a great job of conveying that the IVR knows you were trying to answer, but needs you to try it again. No apologizing needed. The trick here of course is coaching your voice talent to deliver this line in an effective and believable manner. Changing the prosody is a quick and effective way to reprompt, especially in situations where you think the caller was trying to respond correctly, but had some kind of restart/self-correcting/recognition error where no extra information is needed, just another opportunity to try.

Other types of errors are best handled by a strategy of providing more information. The caller may be better served by a prompt that gives tips on how to say an account number (“...like this, three four two et cetera”) or where to find it (“...under your name on your monthly statement.”)

Explicit mention of DTMF options is certainly appropriate to remedy situations of restarts, out-of-grammar responses, or ongoing background noise. Alternatives to speech when there’s the potential for privacy issues—say, with account numbers or medical procedure menus—shows an understanding for how and where the IVR is used.

Then we need to consider if the caller may reasonably not know the information. Depending on the type of application and where its phone number is published, callers may not have their account number in front of them. In this case, they need an out.

For our simple account number prompt, we’ve now come up with several possible reprompts. Here they are all together with an explanation of where each would be appropriate to use.

| Error | Prompt variant |
|---|--|
| No match: caller restarted, was distracted, some digits detected, or prefiller “My account number is...” recognized. | Whaat was that account number? |
| No input: caller typically has a statement in front of them and just needs help finding the account number on it | Please give me your account number. It's in the blue box in the top left corner of your statement. |
| No match with low confidence: caller is saying the account number digits as natural numbers, e.g. “fifty-one thirteen” because they’re printed in groups of two on the statement | Please say or enter your account number. For example, three four two et cetera. |
| No match or no input: company knows a small number of callers don't know or don't yet have an account (Note: if it's a large number, this should be included in an initial prompt to prevent errors.) | Say or enter your account number, or say 'I don't have it.' |

Utilize the grammars

Now let's look at how errors tie into grammars. If the grammar is flexible as to format, the initial prompt can be format-free, but the reprompt can suggest the easiest to recognize format and limit the grammar. For example, look at a flight information prompt. The initial prompt may be as simple as, "Where are you leaving from?" The grammar accepts "BOS," "Boston," "Boston Massachusetts," "Boston Logan," and "Logan." But if the departure airport cannot be determined on the first attempt, reprompt with more directive instructions like this, "Please tell me the city and state you're leaving from." This does two things: it guides the caller to a specific way to formulate his or her answer, and it aids recognition accuracy: no Austin/Boston confusion when it's followed by "Massachusetts." Something to consider in this case is using a stripped-down grammar that only accepts city/state combinations. Combining the directions with the smaller grammar with less confusability should increase the likelihood of successful recognition.

The flipside is that if a caller originally said "Logan" and was misrecognized, so barges in over the reprompt saying "Logan" again, they now have no chance of being recognized. At this point, if we really want to strip down this grammar, we may have to turn barge-in off for the reprompt to make sure the caller hears what we're saying.

Multi-slot grammars allow for even more possibilities. This is one area with clear guidelines. If the original question fills more than one slot, take whatever you get and fill the others later. If you're asking for a departure date and destination city, write the grammar to accept either or both, adding a follow-up question if necessary. Second, if the recognizer comes back with a complete no match, you can back off the multi-slot prompt and ask for one thing at a time.

We said we were going to focus on no match and no input situations, but allow us an interesting exception. Date collection brings with it endpointing issues. It can be hard for the recognizer to distinguish between "February fifth" and "February fifteenth," triggering a false accept error. Consider this healthcare example. The caller is requesting a claim for service on February 15th. The system comes back and says it couldn't find anything for February 5th. Would the caller like to try another date? If the caller says yes, a reprompt could be written like this, "To help me understand you better, please say the date like this: 'the 12th of April.'" Instructing the caller to put the date before the month eliminates the endpointing issues and increases the likelihood of correct recognition. The caveat is you have to be able to effectively communicate the format to the caller.

Most recognizers don't return what they think was recognized when the confidence is low. But that information may still be of use. Let's go back to our airport example. You might want to change the error handling based on what the caller was trying to say. For example, if the recognizer returns an easily recognized phrase like "Indianapolis Indiana" with low confidence, the caller probably didn't actually say that. Under favorable conditions, "Indianapolis Indiana" is usually recognized with high confidence. It was more likely something out of grammar or background noise that the recognizer matched to something in the grammar. But if it returns "I-N-D" (a harder to recognize phrase) with

(not surprisingly) medium confidence, it's probably a valid response. In this case, you could reprompt to say the city and state instead of just the airport code. You can use what you know about the error to guide your rephrasing instructions.

Along those same lines, passing back pre-filler and post-filler as slots can help. Let's say that in response to our airport prompt, "Where are you leaving from?" the caller says, "I'm leaving from DFW." The grammar will of course accept this pre-filler of "I'm leaving from" based on the prompt. But we as designers tend to throw that away. If instead the grammar was written with a pre-filler slot and an airport slot and written to return either or both, we would know from the successful filling of the first slot that the caller was genuinely attempting to answer the question.

A special case of a no match error occurs at the first prompt of a call. Almost all calls start with some kind of greeting and maybe a monitoring message. Often barge-in is turned off for these and callers may be continuing their side speech while it's going on. Then we turn barge-in on and hit them with the first question. If they're still talking to somebody else, they have completely missed all but the first couple of syllables. They'll quickly realize what's happened. At this point, it's probably best to pretend like nothing happened and simply repeat the first question as originally written.

Conclusion

Hopefully we've given you lots of things to think about when writing error reprompts. The bottom line is that the type of error drives the reprompt. If the caller was trying to play along and just wasn't recognized, a very short reprompt is enough. If they don't know how to respond, then examples or further explanation are warranted. The trick is differentiating between them. And as noted earlier, the more information you have available to you, the more variation you can have in your reprompts. You may have three different reprompts at the same spot depending on what you know about the caller's initial response. At some point there are probably diminishing returns to writing all these variations. The goal after all is to get the caller successfully through the prompt. If a single version can do that for all situations, that's the way to go. You just have to find that magical single version.