LumenVox®
Speech Understood

COLLECTING DATA
FOR SPEECH TUNING

Executive Summary:
When performing speech tuning, it is important to collect relevant data to perform an analysis on. This whitepaper describes why this is the case, talks about the amount of data needed, how to collect that data, and how to collect relevant data instead of irrelevant data.

Audience:
This whitepaper is intended for a general audience. However, while no particular technical knowledge is required to understand the content, its focus is on how to make precise measurements. It may be best suited for technical audiences, or at least those with a more technical bent.

INTRODUCTION

One of the key points LumenVox teaches new speech developers is the need to perform "tuning" on any speech application. However, when developers try to implement this advice, they run into some common problems due to some misconceptions about data collection. This whitepaper will address those misconceptions and explain the best practices to follow when collecting data for speech recognition.

"It didn't understand me on that last call!"

Far and away the most common misconception is that a single bad call or recognition is a reason to tune. It is very important to understand that **all speech applications should be tuned, regardless of how well they "seem" to work**. This is because proper tuning is the only way to really know for sure how well an application is actually working (transcription and data analysis are generally the first steps in any tuning cycle).

Automatic speech recognizers (ASRs) are probability engines that compare incoming audio to internal acoustic and language models to make a hypothesis about what was said. That's a fancy way of saying sometimes they get things wrong. The flip side is sometimes an ASR will get something right even if the accuracy for a grammar is very low.

That you had a bad (or good) experience on a single call does not say very much about the application's overall quality. It could be that you got unlucky (or lucky), or it could be that your intuition is right and the application is not performing well. The truth is until you have done proper tuning, you cannot say for sure.

"I had my neighbor call in and it didn't work for him, either."

Following from the first point, just having a few people call into the system is not acceptable either for proper tuning. While technically the chance of luck (either good or bad) skewing the results does decrease by adding a few more users, it generally does not decrease enough to be meaningful until you have at least dozens, if not hundreds, of users trying the system.

In a sense, collecting data for speech tuning is similar to political polling. If you wanted to conduct a political poll to find out who will win the next election, it is obvious that you couldn't just ask a few friends how they were voting. A good poll requires that you talk to a lot of voters in order to extrapolate how the election will turn out. Note that you don't have to talk to every voter (far from it), but you generally need to talk to at least a few hundred.

What you need is what's called a representative sample. This means that the people you talk to (the sample) must be large enough that the role of luck is minimized, and also that the type of people you talk to are representative of the overall population of voters — if you just talked to members of one political party, you would expect to get skewed results.

The same is true with understanding how well a speech application is working. You must get enough callers and utterances that you minimize the role of variance (luck) while also ensuring that the people in your tuning sample share the characteristics of the people who will call into the actual live system. This means that if 50% of your production users are male and 50% are female, having a tuning set that is 80% male and 20% female would be non-representative.

REQUIREMENTS FOR GOOD SAMPLES

As mentioned above, the two major requirements for a good sample are size and representativeness. Both are important, but size is often the harder thing to get just because it takes a lot of person-hours in order to collect the hundreds or thousands of utterances that are needed to do good tuning. However, it is definitely worth it.

The Bigger, the Better

Calculating the actual number of utterances you need to really tune is a little complicated, but a good rule of thumb is around 1,000 utterances per grammar as a minimum. For smaller grammars this may be more than you need and for larger grammars it may not be enough (e.g. if your grammar has 10,000 names in it than obviously 1,000 utterances will not cover most of the things that can be said), but for most grammars used in IVRs it is a close approximation. Note that while too few utterances is bad, too many isn't a problem except that it takes more time to transcribe and analyze that much data. Bigger in this case is almost always better.

It is possible to measure accuracy and tune with fewer utterances, but any conclusions you draw from those measurements should be considered less certain than if you had more data. A good compromise is to do an initial verification round of tuning with less than 1,000 utterances per grammar. This verification round might be thought of as a simple "sanity test" as it can confirm that the application is roughly working.

Any accuracy results you derive from such a test should have a large margin of error. Calculating the exact margin of error is another complex task beyond the scope of this paper, but it's reasonable to think that with 100 utterances your margin of error might well be +/- 20%. That means if you measured 75% accuracy your true accuracy would probably fall somewhere within the range of 55-95%.

You can see how even with a large margin of error there is still some benefit to this verification. If you had 100 samples and measured only 20% accuracy, there is a very high probability that the system is not functional, since the high range of your accuracy prediction would still only be 40%, which is quite low for most applications.

The purpose of the verification round is just to get comfortable enough with the system to know that it can be deployed — perhaps in a limited public beta — and still be somewhat usable for most users. Your goal should always be to collect more data from this deployment so that you can do a proper cycle of tuning and analysis.

Matching Your Sample to Your Population

The other aspect of good samples is that they are representative, as mentioned before. A good way to get a representative sample is through what's called random sampling, which means you expose the system to real users and randomly select a subset of them to analyze. Assuming you get enough users in your subset, your sample will tend to match the population.

Random sampling is easier said than done, however. A random sample must be truly random, meaning that every user must have an equal chance of being represented in the sample. If you are only collecting data during business hours and there are some users who only call in during the middle of the night, then your sample would not be random (in fact it would be biased against a certain class of users), and thus less useful.

The nice thing about speech tuning compared to political polling is that we can actually just collect everyone who uses the system by leaving data collection turned on for a long period of time, allowing all users an equal chance to be represented in the sample. Again, be careful for pitfalls that might exclude certain types of users: if you have a banking IVR for instance, you would want to be sure that you collect data near paydays (e.g. the first and fifteenth of the month) as well as during slower times.

When trying to build representative samples, the number of users you include plays a part. Ideally, when you were collecting 1,000 different utterances per grammar, each utterance would come from a different user who was randomly selected. This would give you a wide variety of voices that matched the variety of voices in your user base.

This is often not possible, though, so again we recommend a compromise. During your initial verification round of tuning, try to get at least 50 speakers. You may not be able to select them randomly in this case, so you should try and deliberately select a non-random sample that matches your population as closely as possible. This means looking at the following:

- What is the ratio of men and women in the sample compared to the user base?
- Are different accents and regional dialects in your sample different from the proportions of accents in the population? This is a particular concern in the world of American IVR development, because engineers and developers are generally more likely to be non-native US English speakers than the users of a typical IVR.
- Are different age groups represented in the sample?
- Are the noise conditions in your sample similar to the noise conditions you expect from your users?

These four factors — gender, regional accent, age, and noise profile — are four of the most influential items in determining recognition accuracy, so they should be controlled as much as possible.