Improving Recognition in SRGS by Adding Weights
Why Weight?

- Speech recognizers are probability engines.
- Adding weights helps the recognizer when confronted with multiple search paths through a grammar.

This is particularly important with large grammars, vital in doing searching.

Also important when accuracy requirements are high.
How to Weight

- Weighting allows you to take what you know about your users and have that knowledge help the recognizer.

- Thus, you will need to have some idea of how your users will respond to prompts.

- Be careful. Weighting is a powerful tool, which means it can be a dangerous tool.
Prescriptive Approach

- You know going in to the application design that users are more likely to answer based a certain way.

- E.g. asking callers in which city they live for a national phone system, higher population cities are more likely to be picked.

Therefore, “New York” much more likely than “Newark.”
Descriptive Approach

- You have a good sample set of transcribed data from a deployed application. You know for a fact what sort of responses users give, and in what proportion.

- You can also tell exactly what sorts of errors the system is making, and weight appropriately.
Scale

- Recognizers treat weights differently. You will need to experiment and test.
- Weights are relative to one another.
- Weighting is multiplicative, so numbers above 1.0 provide positive weights and below 1.0 negative.
Syntax

- ABNF: Weights are entered in front of grammar items, denoted by front-slashes:

  $\text{rule} = /10/ \text{yes} \mid /1/ \text{no};$

- GrXML: Use the weight attribute in an item element.

  \[
  \begin{align*}
  \text{<one-of>}
  \text{<item weight="10">yes</item>}
  \text{<item weight="1">no</item>}
  \text{</one-of>}
  \end{align*}
  \]
Testing

- It is vital you run tests on any weights you add or change.

- Use a different data set than the one you used to perform the initial weighting.

- Monitor a live system very closely after adding weights.