This paper presents an implementation of a pipeline for segment-level feature tracking and adaptation with Praat. It forms a component of the convergence model framework introduced in Raveh et al. (2017); the goal of the model is to describe and generate phonetic changes as part of human-computer interaction in a spoken dialogue system (SDS) setting. A crucial part of such behavior is the ability to detect, track, manipulate, and finally utilize these changes.

The pipeline was tested with the stimuli used in Gessinger et al. (2017), which were designed specifically to trigger and capture phonetic convergence phenomena. Each feature to be captured by the pipeline is defined as a phoneme sequence. The role of the sequence is to capture the phenomenon in question – be it a single sound, a sound in a specific environment, or a phonological process.

The Process begins with a labeled segment matching one of the defined sequences. This input data can be provided by a speech recognition engine, annotated speech corpus, or even be manual specification. The segment’s attributes (starting and ending points, label(s), etc.) are then sent as parameters to a Praat script designed to measure this feature. Figure 1 shows the pipeline’s flow.

Since independent scripts are used for the different features, new features (or the measurement of existing features) can be easily added and modified. Moreover, no programming knowledge is required for doing so (aside from Praat scripting), so that language engineers can utilize their expertise and experiment with the features in question without being dependent on programmers.

This feature tracking and adaptation pipeline has been integrated into a module of an SDS for adding convergence support in addition to the typical architecture of such a system (see, e.g. Chen et al., 2016). The advantages and disadvantages of this method, as well as technical description of the involved steps will be discussed.
Figure 1: The pipeline’s entire flow. Steps surrounded by an diamond are parameterized, i.e. they are dependent on some values configured by the user, like the allowed range of a feature for the “Verify value range” step. Broken lines are optional transitions between steps that apply only if a condition is met. Praat is used in “Measure target value(s)”, which stands at the core of the process and enables all the subsequent steps.

References

