Chronset: Supplementary Information

Additional details regarding human onset detection

For each data set, speech onset latencies were measured by a group of human raters. In the case of the English dataset, ratings from one human were already available; these were supplemented by data from two additional raters who rated the individual trials after they had been parsed into individual wav files (the original recording was a single continuous recording separated by marker tones). The two additional raters performed this task using the Audacity software by listening to the recording and examining the waveform and spectrogram, whereas the original rater used the SayWhen software. For the Spanish data set, data from two raters were available.

Detailed comparison of onset latencies determined by human raters

A core assumption in the literature is that the most precise way to measure voice onset latency is by having a human rater examine the waveform and/or spectrogram. This has led to the adoption of human ratings as the gold standard against which automatic detection algorithms are compared. Given how well Chronset is able to account for the latency data, as well as the lack of a perfect one-to-one match between different human raters, the degree to which the human ratings should be treated as a gold standard when automated detection algorithms approach asymptotic performance becomes an increasingly important question.

To determine the degree to which the data from human raters should be treated as a gold standard, we conducted a detailed comparison between the performance of two human raters on the English dataset. One of these raters was a professional research assistant with several years of training on tasks that included speech onset detection based on visual inspection of both the waveform and the spectrogram of a recording. The second rater was a new research assistant with no experience with the task. A comparison between these two raters therefore provides insight not only into the level of agreement between raters, but how rating agreement could vary as a function of experience with the task.

Prior to beginning the rating task, the new research assistant was instructed in how to identify onset latencies based on the waveform and spectrogram by the trained research assistant and the second author. Both research assistants then coded the data from the first participant in the dataset. The resulting ratings were inspected by both research assistants and the second author as a group to identify the source of any discrepant ratings and to improve agreement. These data were then discarded and both research assistants coded the data from each participant in order.

Figure S1 plots the latencies identified by each rater, as well as the regression slope that best fit these data. Overall, there was a high level of agreement, although there were some discrepancies that appeared to be more common for some participants than others. To quantify these discrepancies and gain finer-grained insight into the agreement levels between the raters, we extracted the slope, intercept, and degree of fit (r^2) between each rater for each participant, and then plotted how these values changed as a function of experience completing the rating task (here reflected by participant number, with coding starting proceeding across subsequent participant numbers). Figure S2, S3, and S4, plot the agreement levels for each of these parameters. In all cases, the regression parameters were relatively stable, showing little change as a function of experience. This suggests that an initial training session by experienced raters is sufficient for new research assistants to produce high-quality estimates of latency onsets. However, there is one caveat --- whereas the near-one value of the slope and the percent of shared variance indicates that rater agreement was similar for both fast and slow trials, the intercept was offset systematically by 17 ms. This reveals that the raters were biased in the amount of

evidence that was needed before they would indicate that a vocal response had been made. This finding has important ramifications for how automatic algorithms should be evaluated relative to human data. Given that different humans show high levels of overall agreement but, at least in our data, may also be systematically offset, relative agreement is, in some respects at least, a better way of assessing performance because it is not influenced by such bias. For this reason, we argue that a better gold standard is one that is a composite of both raw difference scores and relative scores that can be determined via regression residuals.

Rationale for the threshold method for identifying onsets using multiple acoustic features

In designing Chronset, we considered several different possible methods for how the thresholds could be used to identify onsets before adopting the method reported in the main text. One simple approach that is particularly worthy of discussion was why we did not simply sum the values of each of the different features and establish a single threshold value that, once exceeded, would denote the onset of speech. We did not do so because the difference between speech and noise often occurs in a different range of values for different features. Additionally, the change between speech and nonspeech may be nonlinear, which would make the use of a simple (linear) sum of feature values inappropriate. To address this issue, we established different threshold values for each feature and considered speech to be present in the signal when multiple features were above threshold.

Optimization of thresholds for onset detection

To determine when speech occurs, Chronset requires that a set of thresholds are crossed simultaneously for at least four different acoustic features. To ensure the robustness of the thresholds estimated in the present study, we estimated the thresholds for each individual feature by employing a gradient descent approach which builds on neural network optimization techniques and other related methods. Prior to optimization, the recorded speech waveforms were divided into a training set (consisting of 80% of the total audio files) and a testing set (the remaining 20% of the audio files). Optimization was performed on the training set, whereas the testing set was used to assess the robustness of the features. Feature thresholds were set to an initial set of values: 0.1 for all features except WE and FM, which were set to 0.9, effectively requiring little evidence of a speech-like signal to cross a feature threshold; See Figure 5a. The overall fit provided by the estimated thresholds was calculated by comparing human latencies against the latencies estimated by Chronset using maximum likelihood regression. A poor fit of the estimated thresholds was associated with a higher standard deviation (SD) of differences between human and automatic latencies, whereas a better fit was associated with a lower SD. This measure therefore served as the objective function for discovering threshold combinations that lower the overall SD.

To optimize the individual feature thresholds, we first randomly selected one of the six features and then modified this feature by adding and subtracting a feature-specific value *vf* (initial value of 0.05) from the current threshold specific to that feature to produce two new sets of features (one for the addition and one for the subtraction of *vf*). We then compared the SD associated with the new thresholds to that obtained on the basis of the initial values of the thresholds. If one of the modified threshold sets yielded a lower SD than the thresholds prior to modification, the default set of thresholds was updated with this new set; otherwise, the default set of thresholds was retained. To accelerate the gradient descent process, if the default set was changed the value of *vf* was also increased by multiplying the current value by a rate increment of 1.1. This ensured that the optimization was able to move away from poor threshold values rapidly. In contrast, if the default set was not changed this could be due to the target threshold being close to an optimal value, such that a large increase or decrease made the overall fit of the data worse. For this reason, if the default threshold did not change, *vf* was decreased by multiplying the current value by a rate decrement of 0.5 (see Figure 5c for the values of *vf* as a function of the number of attempted feature updates, which we refer to as *iterations*). Note that rates decreased faster than they increased because smaller

threshold changes in the correct direction will always improve the set, albeit slowly, whereas larger threshold changes do not improve the set.

The optimization algorithm attempted 1000 iterations of updating the threshold values before stopping, or stopped after 50 successive iterations had failed to alter the thresholds. Inspection of SD as a function of total number of iterations indicated that a low SD plateau had been reached at this point and the values of *vf* for each of the thresholds were very small; both indicators that an optimized set of thresholds had been reached. The overall fit of the thresholds was then measured by comparing the human ratings against automatic latencies for the test data. To maximize the likelihood that the best possible thresholds were identified, we repeated this optimization process for 200 different partitions of training and test trials. The final set of thresholds were those associated with the smallest SD on the testing data (Figure 5d).

S1. Scatterplots of the latencies identified by rater 1 (x-axis) and rater 2 (y-axis) for each participant (identified by their participant code) in the data set.

Figure S2. Slope of the regression between the raters as a function of experience

Figure S3. Intercept of the regression between the raters as a function of experience.

Figure S4. Percent of variance shared (r_2) across the ratings of both raters as a function of experience.

Figure S5. [upper left] Plot of the standard deviation from the MLE for the training data (black) and test data (red) as a function of number of iterations in the optimization. [upper right]. MLE distribution for the test data set. [lower left]. Size of the change (increment & decrement) for each feature threshold as a function of number of iterations in the optimization. [lower right]. Plot of the initial feature thresholds (black) and optimized feature thresholds (red).