What is going on in someone's head when they do melodic dictation?

Aubrey Hickman Award Application

David John Baker

Levels of Abstraction

In his 2007 article *Models of Music Similarity*, Geraint Wiggins distinguishes between *descriptive* and *explanatory* models in describing the modeling of human behavior (Wiggins 2007). Descriptive models assert what will happen in response to an event. For example, as the note density of a melody increases and the tonalness of a melody decreases, a melody may become harder to dictate (Baker, Monzingo, and Shanahan 2018). While the increase in note density is assumed to drive the decrease in dictation scores, merely stating that there is an established relationship between one variable and the other says nothing about the inner workings of this process. An explanatory model on the other hand not only describes what will happen, but additionally notes why and how this process occurs. For example, work musical expectation demonstrates that as an individual's exposure to a musical style increases, so does their ability to predict specific events within a given musical texture (Pearce 2018).

Not only does more exposure predict more accurate responses, but many of these models of musical expectation derive their underlying predictive power from the brain's ability to implicitly track statistical regularities in musical perception (Saffran et al. 1999; Margulis 2014). The *how* derives from the tracking of statistical regularities in musical information and the *why* derives from evolutionary demands; Organisms that are able to make more accurate predictions about their environment are more likely to survive and pass on their genes (Huron 2006).

Wiggins writes that although there can be both explanatory and descriptive theories, depending on the level of abstraction, a theory may be explanatory at one level, yet descriptive at another. Using the mind-brain dichotomy, he asserts that the example of a theory of musical expectation could be explanatory at the level of behavior as noted above, but says nothing about what is happening at the neural level. Both descriptive and explanatory theories are needed: descriptive theories are used to test explanatory theories and by stringing together different layers of abstraction, we can arrive at a better understanding of how the world works.

One area of research lacking in any explanatory models is work on melodic dictation. Teaching melodic dictation involves instructing students on what and where to direct their attention in order to improve their abilities. This process has been formalized by Gary Karpinski into four discrete steps of hearing, memorizing, understanding, and notating, which help students break down the overwhelming amount of mental processes they need to coordinate in order to successfully complete a melodic dictation (Karpinski 2000). As students' experience increases, they are able memorize larger chunks of music and more easily able to dictate music they once found difficult. Under Wiggins' framework the Karpinski model of melodic dictation (Karpinski 2000, 1990) qualifies as a descriptive model. The model says what happens over the time course of a melodic dictation– specifying four discrete stages discussed in earlier chapters– but does not explicitly state *how* or *why* this process happens. But what is going on in the student's minds over the course of aural skills instructions that allows for this growth? In order to have a more complete understanding of melodic dictation, an explanatory model is needed.

This paper puts forward a computational, cognitive model of melodic dictation with the goal of helping explain how students become better at melodic dictation. The model is based in research from both cognitive psychology (Cowan 2010) and computational musicology (Pearce 2005, 2018) and incorporates relevant theoretical aspects such as working memory and the structure of the melody itself that contribute to a student's performance. In this paper I demonstrate how modeling the cognitive decision process during melodic dictation helps provide a precise framework for pedagogues to understand the inner workings cognition during melodic dictation and can help inform teaching practice.

In addition to quantifying each step, the model incorporates flexible parameters that could be adjusted in order to accommodate individual differences, while still relying on a domain general process. By relying on cognitive mechanisms based in statistical learning, rather than a rule based system for music analysis (Lerdahl and Jackendoff 1986; Narmour 1990, 1992; Temperley 2004) this model allows for the heterogeneity of musical experience among a diversity of music listeners.

Presenting a computational model additionally demonstrates every ontological commitment, thus making it completely vulnerable to criticism allowing it serve as a point of conversational departure in discussions of best practice for melodic dictation pedagogy. This paper directly address the recurring call (Butler 1997; Karpinski 2000; Klonoski 2006) to address the chasm in research between music cognition and music theory pedagogy.

Model Overview

The model consists of three main modules, each with its own set of parameters:

1. Prior Knowledge
2. Selective Attention
3. Transcription and Re-entry

Inspired by Bayesian computational modeling, the *Prior Knowledge* module reflects the previous knowledge an individual brings to the melodic dictation. The *Selective Attention*– somewhat akin to Karpinski's extractive listening– segments incoming musical information by using the window of attention as conceptualized as the limits of working memory capacity as a sensory bottleneck to constrict the size of musical chunk that an individual could to transcribe. Once musical material is in the focus of attention, the *Transcription* function pattern matches against the *Prior Knowledge's* corpus of information in order to find a match of explicitly known musical information. The *Transcription* function will recursively truncate what musical information is in *Selective Attention* if no match is found. In addition to *Transcription*, there is also a *Re-entry* function that will restart the entire loop. This process reflects, but does not actually mirror the exact cognitive process used in melodic dictation, yet seems to be phenomenologically similar to the decision making process used when attempting notate novel melodies. Based on both the prior knowledge and individual differences of the individual, the model will scale in ability, with the general retrieval mechanisms in place. The exact details of the assumptions, parameters, and complete formula of the model are discussed below.

Contents of the Prior Knowledge

The *Prior Knowledge* consists of a corpus of digitally represented melodies taken to reflect the implicitly understood structural patterns in a musical style that the listener has been exposed to. The logic of representing an individual's prior knowledge follows the assumptions of both the Statistical Learning Hypothesis (SLH) and the Probabilistic Prediction Hypothesis (PPH), both core theoretical assumptions of the Information Dynamic of Music (IDyOM) model of Marcus Pearce (Pearce 2005, 2018). Using a corpus of melodies to represent an individual's prior knowledge relies on the Statistical Learning Hypothesis which states:

> musical enculturation is a process of implicit statistical learning in which listeners progressively acquire internal models of the statistical and structural regularities present in the musical styles to which they are exposed, over short (e.g., an individual piece of music) and long time scales (e.g., an entire lifetime of listening). p.2 (Pearce, 2018)

The logic here is that the more an individual is exposed musical material, the more they will implicitly understand it which leads the corroborating probabilistic prediction hypothesis which states:

> while listening to new music, an enculturated listener applies models learned via the SLH to generate probabilistic predictions that enable them to organize and process their mental representations of the music and generate culturally appropriate responses. p.2 (Pearce, 2018).

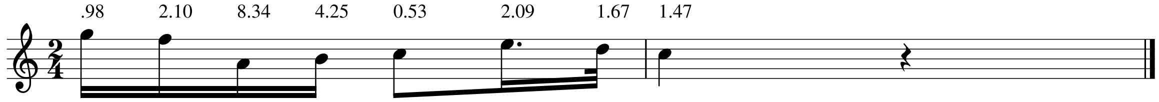


Figure 1: Cadential Excerpt from Schubert's Octet in F Major

Taken together and then quantified using Shannon information content (Shannon 1948), it then becomes possible using the IDyOM framework to have a quantifiable measure that reliably predicts the amount of perceived unexpectedness in a musical melody that can change pending on the musical corpus that the model is trained on. As a model IDyOM has been successful mirroring human behavior in melodies in various styles (Pearce 2018), harmony– outperforming (Harrison and Pearce 2018) sensory models of harmony (Bigand et al. 2014)–, and is also being developed to handle polyphonic materials (Sauve 2017).

Stepping beyond the assumptions of IDyOM, the prior knowledge also needs to have a implicit/explicitly known parameter which indicates whether or not an pattern of music– or n-gram[1] pattern– is explicitly learned. This threshold can be set relative to the entire distribution of all n-grams in the corpus.

Having established that the models' first parameters to be decided are the representation of strings and the implicit/explicit threshold, the next decision that has to be made is how the model decides segmentation for the second stage of *Selective Attention*. Although there has been a large amount of work on different ways to segment the musical surface using rule based methods (Lerdahl and Jackendoff 1986; Margulis 2005; Narmour 1990, 1992), which rely on matching a music theorist's intuition with a set of descriptive rules somewhat like the boundary formation rules put forward in *A Generative Theory of Tonal Music*, as noted by Pearce (Pearce 2018), rule based models often fail at when applied to music outside the Western art music canon. Additionally, since melodic dictation is an active memory process, rather than a semi-passive process of listening, this model needs to be able to quantify musical information on two conditions. The first is that it must be dependent on prior musical experience. The second is that it should allow for a movable boundary for selective attention so that musical information that is memory can be actively maintained while carrying out another cognitive process, that of notating the melody. In order to create this metric, I rely on IDyOM's use of information content (Shannon 1948) which quantifies the information content of melodies based on corpus of materials.

For example, when trained against a corpus of melodies, this excerpt in Figure 7.1 from the fourth movement of Schubert's *Octet in F Major* (D.803) lists the information content of the excerpt calculated for each note atop the notation[2] Appearing in Figure 7.2, I plot the cumulative information content of the melody, along with both an arbitrary threshold for the limits of working memory capacity and where the subsequent segmentation boundary for musical material to be put in the *Selective Attention* buffer would be. These values chosen show a small example of how the *Selective Attention* module works. The advantage of operationalizing how an individual hears a melody like this is that melodies with lower information content, derived from an understanding of having more predictable patterns from the corpus, will allow for larger chunks to be put inside of the selective attention buffer. Additionally, individuals with higher working memory capacity would be able to take in more musical information.

It is important to highlight that the notes above the melody here are dependent on what is current in the *Prior Knowledge* module. A corpus of *Prior Knowledge* with less melodies would lead to higher information content measures for each set of notes, while a prior knowledge that has extensive tracking of the patterns

[1] n-grams refer to the amount of musical objects in a string. For example a bi-gram or 2-gram, would be an interval. Tri-grams or 3-grams would consist of two intervals and so on.

[2] The following musical examples is taken from Pearce (2018) reflects a model where IDyOM was configured to predict pitch with an attribute linking melodic pitch interval and chromatic scale degree (pitch and scale degree) using both the short-term and long-term models, the latter trained on 903 folk songs and chorales (data sets 1, 2, and 9 from table 4.1 in (Schaffrath 1995) comprising 50,867 notes.

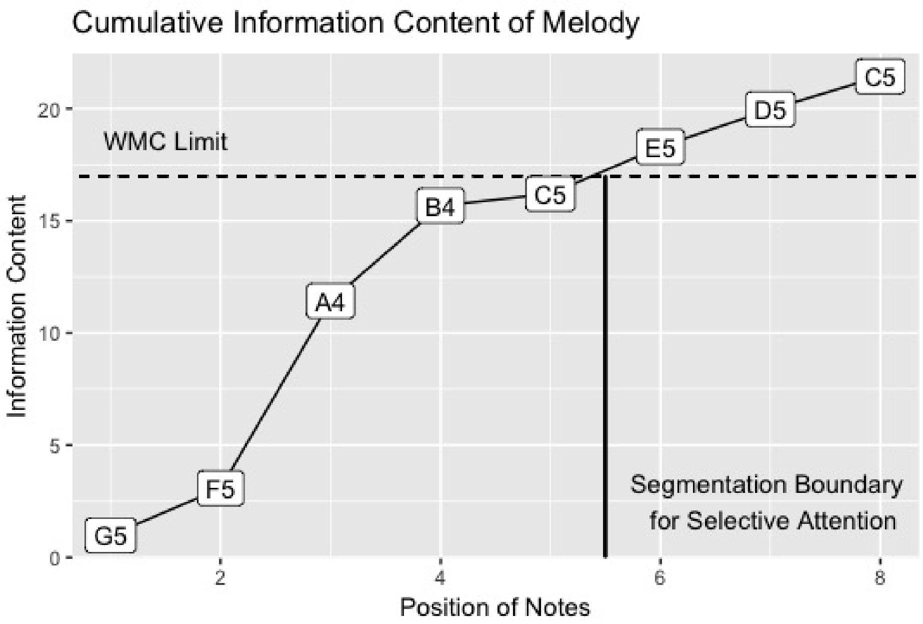


Figure 2: Cumulative Information in Schubert Octet Excerpt

would lead to lower information content. This increase in predictive accuracy mathematically reflects the intuition that those with more listening experience can process greater chunks of musical information.

Setting Limits with Transcribe

With each note then quantified with a measure of information content, it then becomes possible to set a limit on the maximum amount of information that the individual would be able to hold in memory as defined by the *Selective Attention* module. A higher threshold would allow for more musical material to be put in the attentional buffer, and a lower threshold would restrict the amount of information held in an attentional buffer. By putting a threshold on this value, this serves as something akin to a perceptual bottleneck based on the assumption that there is a capacity limit to that of working memory (Cowan 1988, 2010). Modulating this boundary will help provide insights into the degree to which melodic material can be retained between high and low working memory span individuals.

Pattern Matching

With subset of notes of the melody represented in the attentional buffer, whether or not the melody becomes notated depends on whether or not the melody or string in the buffer can be matched with a string that is explicitly known in the corpus. Mirroring a search pattern akin to Cowan's Embedded Process model (Cowan 1988, 2010), the individual would search across their long term memory, or *Prior Knowledge* for anything close to or resembling the pattern in the *Selective Attention* buffer. Cowan's model differs from other more module based models of working memory like those of Baddeley and Hitch (1974) by positing that working memory should be conceptualized as a small window of conscious attention. As an individual directs their attention to concepts represented in their long term memory, they can only spotlight a finite amount of information where categorical information regarding what is in the window of attention not far from retrieval.

When searching for a pattern match, the *Transcription* module is at work. If a pattern match that has been moved to *Selective Attention* is immediately found, the contents of *Selective Attention* would be considered to be notated. The model would register that a loop had taken place and document the n-gram match. Of course, finding an immediate pattern match each time is highly unlikely and the model needs to be able to compensate if that happens.

If a pattern is not found in the initial search that is *explicitly* known, one token of the n-gram would be dropped off the string and the search would happen again. This recursive search would happen until an explicit long term memory match is made. Like humans taking melodic dictation, the computer would have the best luck finding patterns that fall within the largest density of a corpus of intervals distribution. Additionally, like students performing a dictation, if a student does not explicitly know an interval, or a 2-gram, the dictation would not be able to be completed. If this happens, both the model and student would have to move on to the next segment via the *Re-entry* function.

Eventually there would be a successful explicit match of a string in the *Transcription* module and that section of the melody would be considered to be dictated. The model here would register that one iteration of the function has been run and the chunk transcribed would then be recorded. After recording this history, the process would happen again starting at either the next note from where the model left off, the note in the entire string with the lowest information content, or n-gram left in the melody with that is most represented in the corpus. This parameter is defined before the model is run and the question of dictation re-entry certainly warrants further research and investigation.

Upon the successful pattern match of a string, the *Selective Attention* and *Transcription* module would need too then be run again. This process is done via the *Re-entry* function.

Completion

Given the recursive nature of this process, if all 2-grams are explicitly represented in the *Prior Knowledge* then the target melody should be transcribed. If only represented using such a small chunk, the model will have to loop over the melody many times, thus indicating that the transcriber had a high degree of difficulty dictating the melody. If there is a gap in explicit knowledge in the prior knowledge, only patches of the melody will be recorded and the melody will not be recorded in its entirety. An easier transcription will result in less iterations of the model with larger chunks. Though the current instantiation of the model does not incorporate how multiple hearings might change how a melody is dictated, one could constrain the process to only allow a certain number of iterations to reflect this. Of course as a new melody is learned it is slowly being introduced into long term memory and could be completely be capable of being represented in long term memory without being explicitly notated at the end of a dictation with time running out and thus not possible to be completed. This of course then would be imposing some sort of experimental constraint on the process and since this is meant to be a cognitive computational model of melodic dictation this caveat would complicate the model. Future research could be done to optimize the choices that the model makes in order to satisfy whatever constraints are imposed and could be an interesting avenue of future research, but are beyond the initial goals of the model.

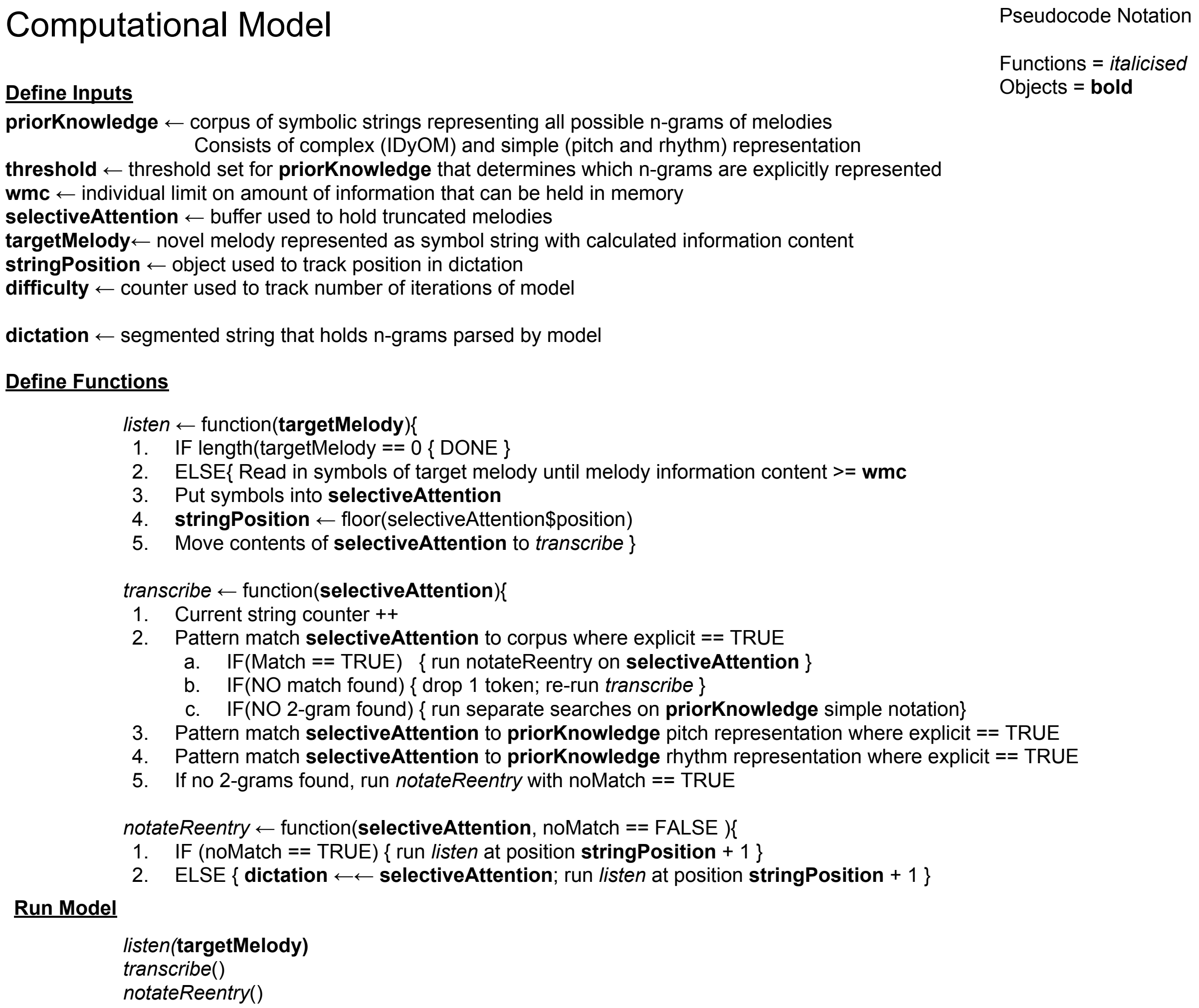
Formal Model

Below I present the computational model in psudeocode as described in Figure 7.3. First listed are the defined inputs, the functions needed to run the algorithm, and then the sequence the model runs. To aid distinguishing between functions and objects, I put functions in italics and objects in bold. Below the model in Figure 7.4, I provide a brief walk through of one iteration of the model.

Computational Model

Example

The example above shows one iteration of the model run using the musical example from above using a hypothetical corpus for the pattern matching. Using the model above, the following inputs were defined *a*



Computational Model

Pseudocode Notation

Functions = *italicised*
Objects = **bold**

Define Inputs

priorKnowledge ← corpus of symbolic strings representing all possible n-grams of melodies
Consists of complex (IDyOM) and simple (pitch and rhythm) representation
threshold ← threshold set for **priorKnowledge** that determines which n-grams are explicitly represented
wmc ← individual limit on amount of information that can be held in memory
selectiveAttention ← buffer used to hold truncated melodies
targetMelody← novel melody represented as symbol string with calculated information content
stringPosition ← object used to track position in dictation
difficulty ← counter used to track number of iterations of model

dictation ← segmented string that holds n-grams parsed by model

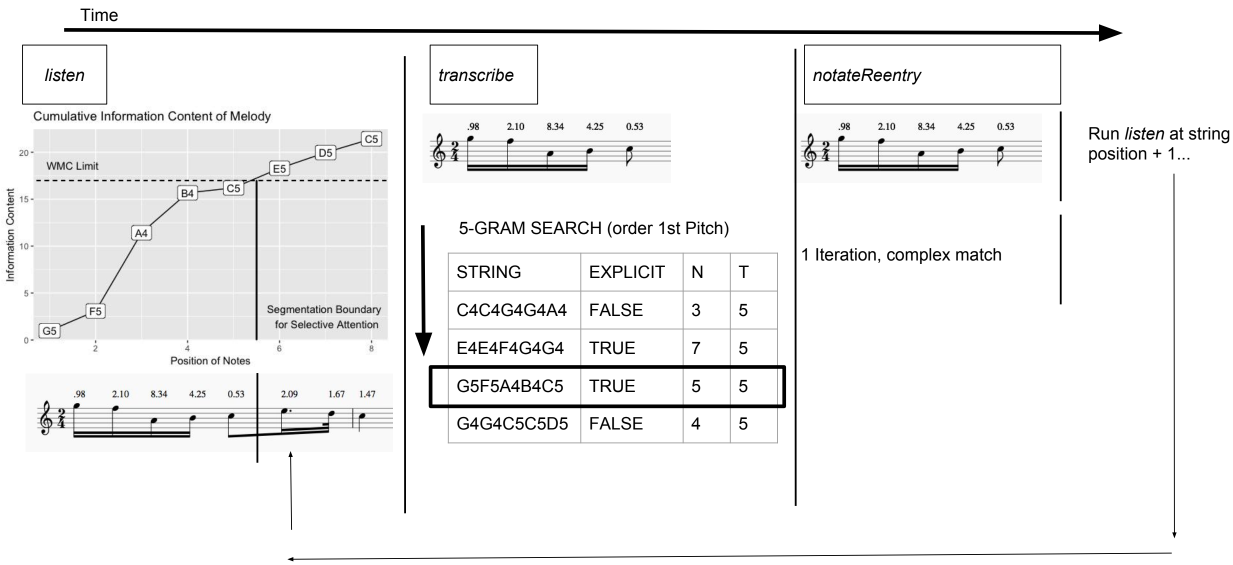
Define Functions

```
listen ← function(targetMelody){
 1.  IF length(targetMelody == 0 { DONE }
 2.  ELSE{ Read in symbols of target melody until melody information content >= wmc
 3.  Put symbols into selectiveAttention
 4.  stringPosition ← floor(selectiveAttention$position)
 5.  Move contents of selectiveAttention to transcribe }

transcribe ← function(selectiveAttention){
 1.  Current string counter ++
 2.  Pattern match selectiveAttention to corpus where explicit == TRUE
     a.  IF(Match == TRUE)   { run notateReentry on selectiveAttention }
     b.  IF(NO match found) { drop 1 token; re-run transcribe }
     c.  IF(NO 2-gram found) { run separate searches on priorKnowledge simple notation}
 3.  Pattern match selectiveAttention to priorKnowledge pitch representation where explicit == TRUE
 4.  Pattern match selectiveAttention to priorKnowledge rhythm representation where explicit == TRUE
 5.  If no 2-grams found, run notateReentry with noMatch == TRUE

notateReentry ← function(selectiveAttention, noMatch == FALSE ){
 1.  IF (noMatch == TRUE) { run listen at position stringPosition + 1 }
 2.  ELSE { dictation ←← selectiveAttention; run listen at position stringPosition + 1 }
```

Run Model

```
listen(targetMelody)
transcribe()
notateReentry()
```

Figure 3: Formal Model



STRING	EXPLICIT	N	T
C4C4G4G4A4	FALSE	3	5
E4E4F4G4G4	TRUE	7	5
G5F5A4B4C5	TRUE	5	5
G4G4C5C5D5	FALSE	4	5

IC calculations based on Pearce (2018)

Figure 4: Model Example

priori:

- The **Prior Knowledge** is a hypothetical corpus of symbolic strings representing all n-grams of melodies
- The **Threshold** is set to **five** exact matches in the **Prior Knowledge**
- The **WMC** is set at 17
- The **Target Melody** is the Schubert excerpt from above
- The **String Position** object is used to track the position in the dictation
- The **Difficulty** object starts at 0
- The **Dictation** object is `NULL` to begin, and each new n-gram successfully transcribed is annexed to it

Figure 7.4 progresses from left to right over the course of time. The algorithm begins by first running the `listen()` function on the **Target Melody**. First the model checks that there are notes to transcribe; this being the first loop of the model, this is statement will be `FALSE` so the next step is taken. Notes of the **Target Melody** are read in to the **Selective Attention** buffer until the information content of the melody exceeds that of the working memory threshold. This is depicted graphically in the leftmost panel of Figure 7.4. Each note unfolding over time fills up the **Selective Attention** working memory buffer. When the amount of information reaches the perceptual bottleneck– as indicated by the dashed line– the **Selective Attention** buffer stops receiving information. At this point the model will mark where in the melody it stopped taking in new information for later. Here the contents in **Selective Attention** are moved to the `transcribe()` function.

With the contents of **Selective Attention** passed to `transcribe()`, the model adds one to the counter indicating the first search is about to run. Moving to the middle panel of Figure 7.4, the symbol string of notes in the first column are indexed against the **Prior Knowledge**. Only if a five note pattern has appeared more than or equal to five times, as determined by the **Threshold** input, will the corresponding `EXPLICIT` column be `TRUE`. In this case, this pattern has occurred over the threshold of 5 and thus a successful match is found. It is at this step that the search resembles that of Cowan's model of working memory as active attention. The pattern being searched for is compared against a vast amount of information, with cues from the contents of what is in **Selective Attention** grouping similar patterns together. At the neural level, this is most likely a much more complex process, but to show this grouping I note that this search is at least organized by the first pitch. I assume it would be reasonable that patterns starting on G as $\hat{5}$[3] might happen

[3] As determined by being calculated against the corpus with both pitch and scale degree information

together. Since this string does have a `TRUE` match with `EXPLICIT`, the contents of **Selective Attention** are considered notated. At this point the model would record the 5-gram, along with the string that it was matched with. the function would then re-run the `listen` function via the `notateReentry()` function at the next point in the melody as tracked by the **String Position** object.

If there were not to have been an exact match, the model would remove one token from the melody and performed the search again on the knowledge of all 4-grams and add one to the **Difficulty** counter. This process would happen recursively until a match is found. If no match is found in either the complex representation, or that of the two rhythm and pitch corpora, the fifth step of `transcribe()` would trigger `notateReentry()` to be run without documenting the n-gram currently being dictated. This would be akin to a student not being able to identify a difficult interval, thus having to restart the melody at a new position. Decisions about re-entry warrant further research and discussion, but this model for the sake of parsimony, assumes linear continuation. As notated in §7.3.5, other modes of re-entry could be incorporated into the model.

This looping process would occur again and again until the entire melody is notated. With each iteration of each n-gram notated, the difficulty counter would increase in relation to the representation of that string in the corpus. This provides an algorithmic implementation of a theorist's intuition that less common n-grams or intervals (2-grams) are going to lead to higher difficulty in dictation. Also worth noting is steps 3 and 4 in the `transcribe()` function are akin to Karpinski's proto-notation. Further research might consider advantages in the order of searching the **Prior Knowledge** corpora.

Baddeley, Alan D., and Graham Hitch. 1974. "Working Memory." In *Psychology of Learning and Motivation*, 8:47–89. Elsevier. https://doi.org/10.1016/S0079-7421(08)60452-1.

Baker, David John, Monzingo Elizabeth, and Daniel Shanahan. 2018. *Modeling Aural Skills Dictation.* Lecture Notes in Computer Science. Graz, Austria: Centre for Systematic Musicology, University of Graz.

Bigand, Emmanuel, Charles DelbÃ©, BÃ©nÃ©dicte Poulin-Charronnat, Marc Leman, and Barbara Tillmann. 2014. "Empirical Evidence for Musical Syntax Processing? Computer Simulations Reveal the Contribution of Auditory Short-Term Memory." *Frontiers in Systems Neuroscience* 8 (June). https://doi.org/10.3389/fnsys.2014.00094.

Butler, David. 1997. "Why the Gulf Between Music Perception Research and Aural Training?" *Bulletin of the Council for Research in Music Education*, no. 132.

Cowan, Nelson. 1988. "Evolving Conceptions of Memory Storage, Selective Attention, and Their Mutual Constraints Within the Human Information-Processing System." *Psychological Bulletin* 104 (2): 163–91.

———. 2010. "The Magical Mystery Four: How Is Working Memory Capacity Limited, and Why?" *Current Directions in Psychological Science* 19 (1): 51–57. https://doi.org/10.1177/0963721409359277.

Harrison, Peter M . C., and Marcus Thomas Pearce. 2018. "Dissociating Sensory and Cognitive Theories of Harmony Perception Through Computational Modeling." https://doi.org/10.31234/osf.io/wgjyv.

Huron, David. 2006. *Sweet Anticipation.* MIT Press.

Karpinski, Gary. 1990. "A Model for Music Perception and Its Implications in Melodic Dictation." *Journal of Music Theory Pedagogy* 4 (1): 191–229.

Karpinski, Gary Steven. 2000. *Aural Skills Acquisition: The Development of Listening, Reading, and Performing Skills in College-Level Musicians.* Oxford University Press.

Klonoski, Edward. 2006. "Improving Dictation as an Aural-Skills Instructional Tool," 6.

Lerdahl, Fred, and Ray Jackendoff. 1986. *A Generative Theory of Tonal Music.* Cambridge: MIT Press.

Margulis, Elizabeth Hellmuth. 2005. "A Model of Melodic Expectation." *Music Perception: An Interdisciplinary Journal* 22 (4): 663–714. https://doi.org/10.1525/mp.2005.22.4.663.

———. 2014. *On Repeat: How Music Plays the Mind.* Oxford: Oxford University Press.

Narmour, Eugene. 1990. *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model.* Chicago: University of Chicago Press.

———. 1992. *The Analysis and Cognition of Melodic Complexity: The Implication-Realization Model.* University of Chicago Press.

Pearce, Marcus. 2005. "The Construction and Evaluation of Statistical Models of Melodic Structure in Music Perception and Composition." PhD thesis, Department of Computer Science: City University of London.

Pearce, Marcus T. 2018. "Statistical Learning and Probabilistic Prediction in Music Cognition: Mechanisms of Stylistic Enculturation: Enculturation: Statistical Learning and Prediction." *Annals of the New York Academy of Sciences* 1423 (1): 378–95. https://doi.org/10.1111/nyas.13654.

Saffran, Jenny R, Elizabeth K Johnson, Richard N Aslin, and Elissa L Newport. 1999. "Statistical Learning of Tone Sequences by Human Infants and Adults." *Cognition* 70 (1): 27–52. https://doi.org/10.1016/S0010-0277(98)00075-4.

Sauve, Sarah. 2017. "Prediction in Polyphony: Modelling Auditory Scene Analysis." PhD thesis, Centre for Digital Music: Queen Mary, University of London.

Schaffrath, Helmuth. 1995. "The Essen Folk Song Collection, D. Huron."

Shannon, Claude. 1948. "A Mathematical Theory of Communication." *Bell System Technical Journal* 27: 379–423.

Temperley, David. 2004. *The Cognition of Basic Musical Structures*. Cambridge: MIT Press.

Wiggins, Geraint A. 2007. "Models of Musical Similarity." *Musicae Scientiae* 11 (1_suppl): 315–38. https://doi.org/10.1177/102986490701100112.