An exemplar-based learning model of English sentence intonation*

Una Y. Chow and Stephen J. Winters
University of Calgary

Abstract: Exemplar theory (Johnson 1997) claims that listeners store exemplars of speech that they have experienced in detail, enabling them to categorize new words in memory without speaker normalization. In applying exemplar theory to intonation perception, we have developed a computational model that categorizes English statements and echo questions by comparing how similar a sentence is to previously encountered sentences, based on three weighted properties: 1) the *timing* of the nuclear tone, 2) the *direction* of the post-nuclear F0 change, and 3) the *speed* at which the post-nuclear F0 changes. When trained and tested on 64 pairs of statements and echo questions, the model correctly categorized over 92% of the sentences when only *direction* was included in the similarity calculation. This result demonstrates that it is feasible to develop an exemplar-based computational model that can learn to categorize English statements and echo questions without normalizing F0 to account for speaker variability.

Keywords: sentence intonation, speech perception, exemplar theory, computational model

1 Introduction

The motivation for this research project was to better understand how listeners can perceive the intonation of statements and questions, given the inherent variability in speech (e.g., Fant 1972; Peterson & Barney 1952; Simpson 2009). In speech, variation in pitch can be used to distinguish between different types of sentences. For example, English speakers typically lower the pitch at the end of a statement (e.g., 'John reads books.'), but raise the pitch at the end of a yes/no question (e.g., 'Does John read books?') (Wells 2006). Echo questions (e.g., 'John reads books?') are a type of yes/no question, which has the same rising intonation as the other types of yes/no questions but the same word order as statements.

This project aimed to investigate whether a working computational model of exemplar theory (Johnson 1997) can successfully perceive the difference between statements and echo questions in English. This theory holds that listeners store exemplars of speech that they have experienced in fine phonetic detail in memory. They can then use the phonetic details of these exemplars to categorize new tokens without the need for speaker normalization (Johnson 1997). When a new token is encountered, its similarity with all the exemplars stored in memory is calculated. The new token is then categorized according to which group of exemplars it is most similar to, overall.

Our methodology was first to create an exemplar-based computational model which can learn to categorize English statements and echo questions, based solely on their intonation patterns, and then to test how well this model can categorize statements and echo questions produced by both a male speaker and a female speaker of English, without normalizing fundamental frequency (F0) to account for speaker variability and gender variability. Although researchers have developed exemplar-based models that can identify single words or sounds (e.g., Goldinger 1998; Johnson 1997), few studies have applied this theory to the perception of intonation (e.g., Walsh,

* This project was funded by the University of Calgary's Program for Undergraduate Research Experience (PURE) Award to Una Chow.
Contact info: uchow@ucalgary.ca

Proceedings of the Northwest Linguistics Conference 33.
D. K. E. Reisinger (ed.). Vancouver, BC: UBCWPL, 2019.

Schweitzer, & Schauffer 2013) and, in particular, sentence intonation (e.g., Chow & Winters 2015). The idea is that if a similarity-based calculation module can accurately classify novel sentences at an acceptable rate on the basis of intonation alone, it could be expanded to account for the human perception of intonation more generally.

2 Proposed Intonation Perception Model

2.1 Exemplar-based Classification

The following model of intonation perception of statements and echo questions in English uses an exemplar-based process of categorization to determine the sentence type (statement or question) of a new token. It adopts a simplified version of the algorithm proposed by Johnson (1997) and Nosofsky (1988).

2.1.1 Similarity Calculation

The model categorizes a new token (i.e., a sentence) based on how similar it is, overall, with all of the experienced tokens (or exemplars) in memory as follows. First, the model derives the auditory distance d_{ij} between the new token i and every exemplar j in category C by calculating the Euclidean distance between their auditory properties m, as defined in (1). Each auditory property is weighted by its attention weight w_m; this attention weight can vary from 0% to 100% and reflects the varying degree of attention that the listener may give to the different auditory properties of the token as the listener experiences it. The specific auditory properties that this model uses in the similarity calculation will be described in Section 2.1.2.

(1) Auditory distance: $d_{ij} = [\sum w_m (x_{im} - x_{jm})^2\ ]^{1/2}$

Secondly, the model derives the auditory similarity s_{ij} between token i and exemplar j by applying an exponential decay function to the auditory distance d_{ij} so that the nearest auditory exemplars have more influence than the more distant exemplars in determining the auditory similarity value, as defined in (2).

(2) Auditory similarity: $s_{ij} = e^{-d_{ij}}$

Finally, the model derives the overall similarity S_i between token i and category C by summing all of the auditory similarity values s_{ij} between i and every exemplar j in C, as defined in (3).

(3) Overall similarity: $S_i = \sum s_{ij},\ j \in C$

2.1.2 Auditory Properties

Prototypically in English, statements end with a falling intonation and echo questions end with a rising intonation (Wells 2006). This fall or rise in F0 commonly starts at or near the nuclear tone (i.e., the last prominent, stressed syllable) of the intonational phrase (Pierrehumbert & Hirschberg 1990), as shown in Figure 1. Since the nuclear tone and the post-nuclear intonation are potentially salient cues for sentence type, we derived three auditory properties for the similarity calculation from this tone: 1) the relative *timing* (expressed as a percentage) of the nuclear tone in the intonational phrase, as defined in (4), 2) the *direction* (rising or falling) of the intonational phrase

immediately following the nuclear tone, as defined in (5), and 3) the *speed* (or the absolute slope value) of the intonational phrase immediately following the nuclear tone, as defined in (6).

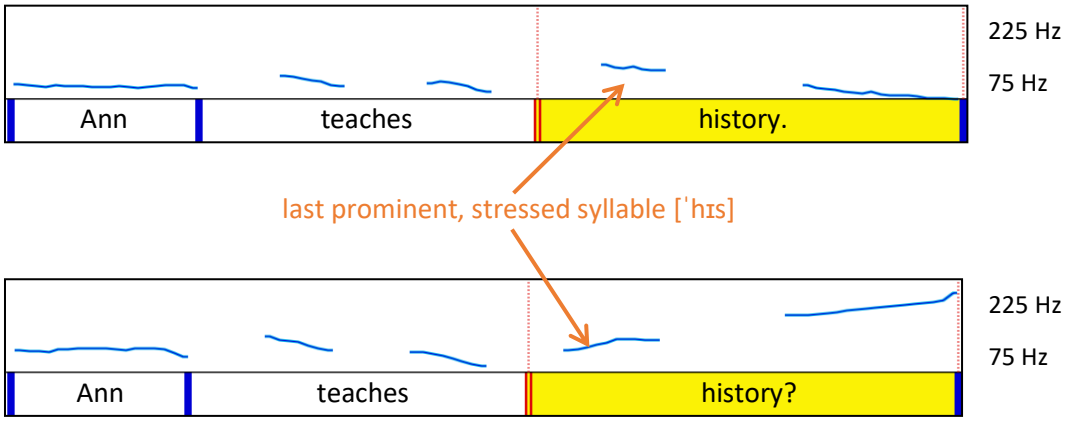

Figure 1: Nuclear tones of the statement and question intonational phrases of 'Ann teaches history' produced by a native speaker of English

(4) Timing: PrenuclearTime / (PrenuclearTime + PostnuclearTime) * 100
where PrenuclearTime = the time in the sentence before the nuclear tone
and PostnuclearTime = the time in the sentence after the nuclear tone.

(5) Direction: If $(y_1 - y_2) / (x_1 - x_2) < 0$, then 'falling';
else if $(y_1 - y_2) / (x_1 - x_2) > 0$, then 'rising';
else 'level'
where (x_1, y_1) = the time and F0 value of the nuclear tone
and (x_2, y_2) = the time and F0 value of a post-nuclear point
in the intonational phrase.

See Section 2.2.2 on the settings of the analysis function of the computational model for more details about how we calculated this post-nuclear point.

(6) Speed: $| (y_1 - y_2) / (x_1 - x_2) |$
where (x_1, y_1) = the time and F0 value of the nuclear tone
and (x_2, y_2) = the time and F0 value of a post-nuclear point
in the intonational phrase.

2.1.3 Categorization Process

Figure 2 illustrates the categorization process for a new token; this token is represented by the F0 contour shown at the top. First, the computational model calculates the auditory similarity between this new token and every exemplar in both the 'question' category (represented by the exemplar cloud on the left) and the 'statement' category (represented by the exemplar cloud on the right). This similarity value is determined by the Euclidean distance of the weighted auditory properties (timing, direction, and speed) between the token and the exemplar. In Figure 2, the

token's auditory similarities with the two question exemplars are 0.7 and 0.9 and the corresponding similarities with the statement exemplars are 0.3 and 0.1. Since the token's overall similarity with the 'question' category (0.7 + 0.9 = 1.6) is greater than that with the 'statement' category (0.3 + 0.1 = 0.4), the model categorizes the token as a 'question'. Once a token has been categorized in memory, it is used, along with the existing exemplars, in the categorization process for subsequent tokens. For instance, for future tokens, the new token at the top of Figure 2 would be treated as an exemplar in the 'question' cloud.

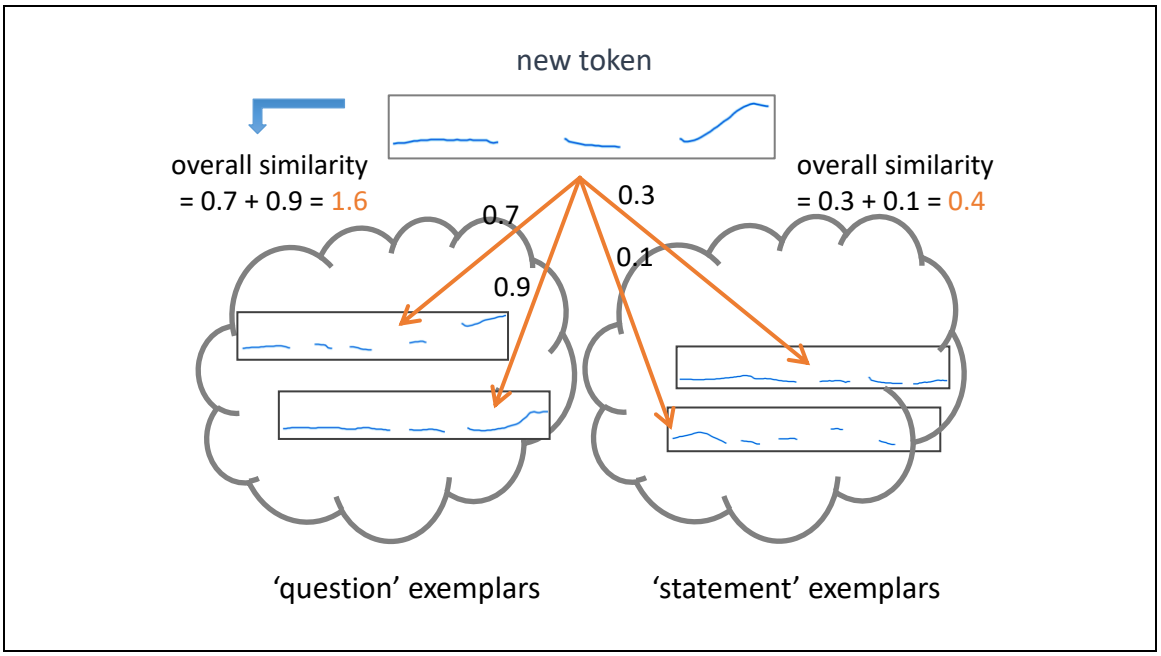

Figure 2: An illustration of the categorization process of a new token

2.2 User-interactive Interface

We designed the model with a user-interactive interface in Praat (Boersma & Weenink 2013) that comprises six functions. As shown on the interface's main menu in Figure 3, the functions are *preanalysis*, *analysis*, *extraction*, *training*, *testing*, and *cross-validation*. These step-by-step functions provide users with the flexibility of adjusting specific settings during each step; they also enable users to view the results in each step and to re-run a step if necessary. To begin, the user specifies the data directory where the sound files are located.

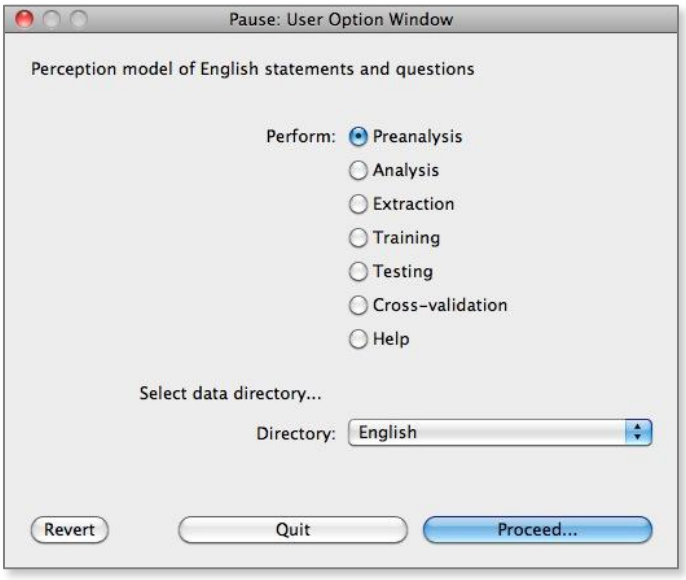

Figure 3: The main menu of the computational model

2.2.1 Preanalysis

The preanalysis function prepares the sound token for acoustic analysis. It reads in the audio-recorded samples of the statements and echo questions from the data directory specified on the main menu and removes any silence (or voiceless part) preceding and following the speech sound. The model's default pitch range for tracking the fundamental frequency of the intonation contour is 75–500 Hz. The user can adjust this pitch range, for example, to 75–600 Hz for analyzing the question intonation of female speakers, which can sometimes go beyond 500 Hz.

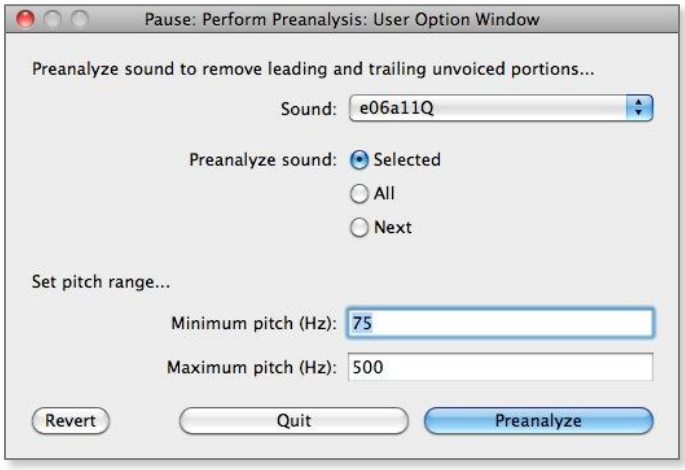

Figure 4: The parameter setting window of the preanalysis function

2.2.2 Analysis

The analysis function locates any salient statement or question cues in the intonation contour of the sound tokens. Often, certain parts of the F0 contour may be voiceless due to voiceless segments (e.g., /h/ and /s/). These voiceless parts appear as gaps in the F0 contour, as shown in the statement and question contours in Figure 5.

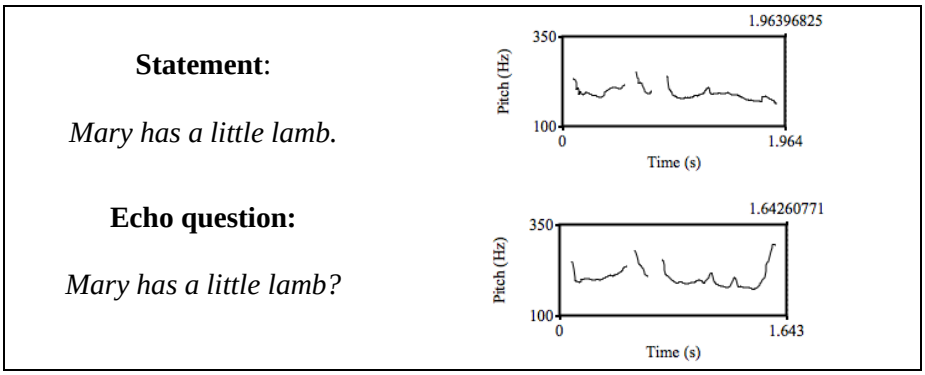

Figure 5: Voiceless gaps in the F0 contours of the statement and echo question: 'Mary has a little lamb'

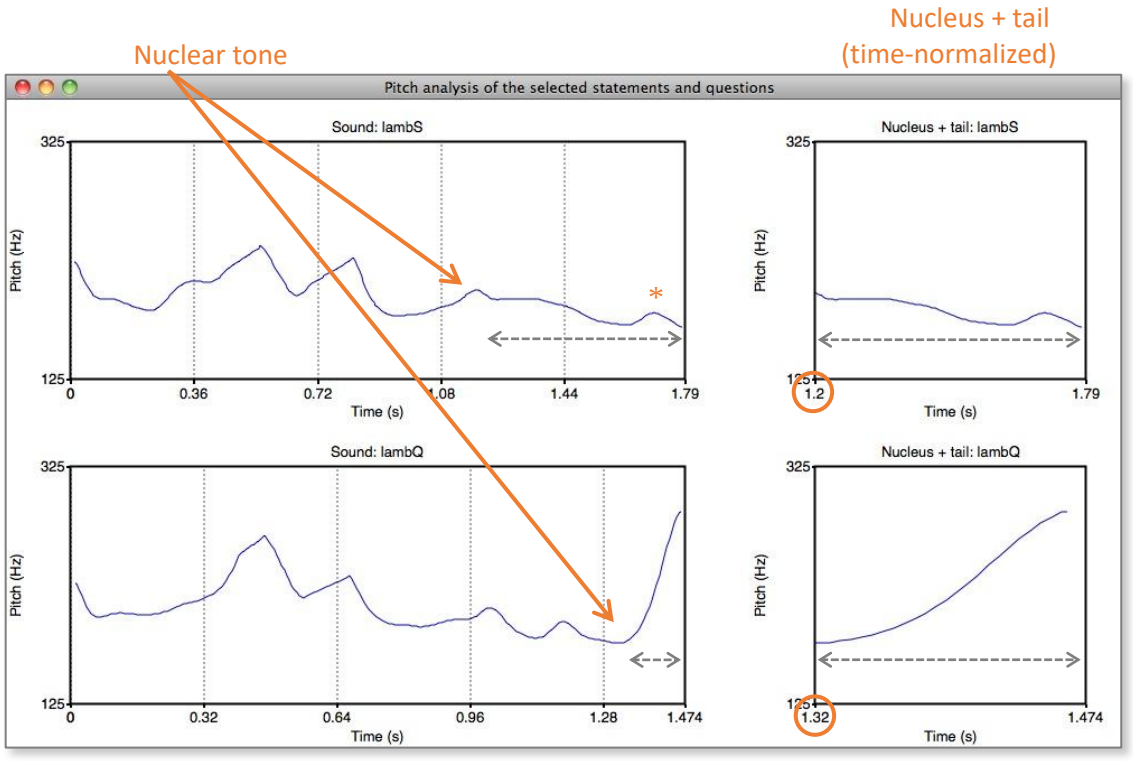

Figure 6: The result of the analysis function, showing the location of the nucleus and the tail of the interpolated statement and question contours of 'Mary has a little lamb'

This function first fills the gaps within an F0 contour using linear interpolation (Praat's interpolate function) in order to create a continuous curve. It then locates the nuclear tone in the sentence contour, as displayed in Figure 6. (More information on how the model accomplishes this is given below.) In Figure 6, the top half shows the F0 contour of the statement and the bottom half shows the echo question of 'Mary has a little lamb'. The boxes on the left show the entire F0 contours of the sentences. The boxes on the right show the tail ends of these F0 contours starting from their nuclear tones. These tails are time-normalized between the two sentence types, as indicated by the grey dotted lines, while their actual durations in time relative to the complete F0 contours are shown in the boxes on the left. In this example, the nuclear tone of the statement occurs at the 1.2 seconds time point of the F0 contour, while the nuclear tone of the question occurs at the 1.32 seconds time point of the F0 contour.

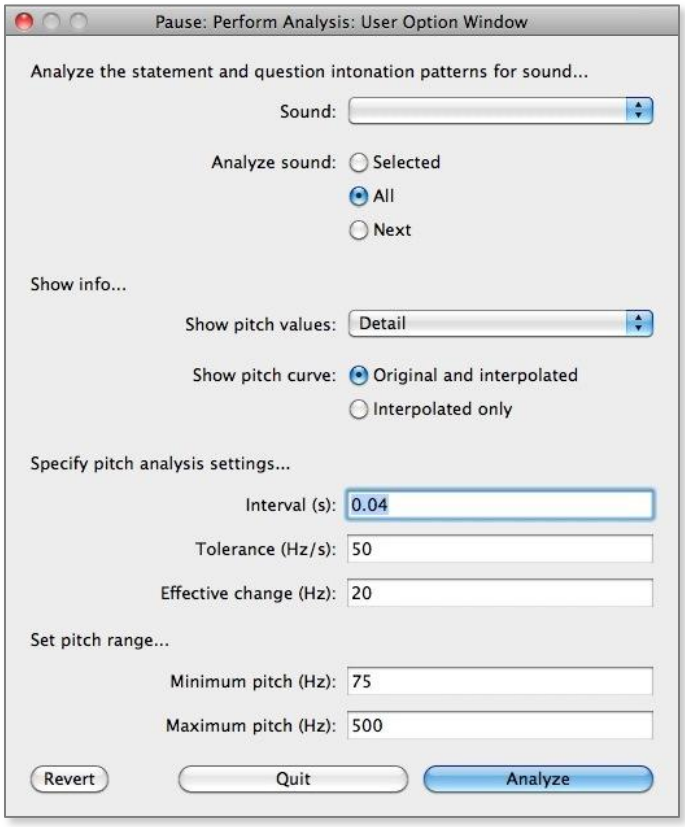

Figure 7: The parameter setting window of the analysis function

To locate the nuclear tone (and the final rise or fall of the F0 contour), the model compares the slopes between successive time points of the F0 contour as follows. First, the model calculates the slope between time point 0 (at time 0 seconds) and time point 1. (The default interval between time points is 40 milliseconds but it can be adjusted in the parameter setting window, as shown in Figure 7.) Then, it calculates the slope between time points 1 and 2 and compares this slope with the first slope to determine whether there has been a change in the direction of the intonation rise or fall. Next, it calculates the slope between time points 2 and 3 and compares it with the second slope. It continues to do so with the subsequent time points until the end of the contour. After this

analysis, the model identifies the nuclear tone as the onset of the final fall or rise in the intonation contour.

Shorter intervals between time points require more computations to determine the time points, slopes, and the changes in slope. On the other hand, longer intervals between time points lead to a greater risk of missing the exact location of the nuclear tone. The user must balance these concerns in setting this parameter in order to yield the most accurate and efficient performance of the model.

Additionally, there may be tiny bumps in the F0 contour that can be mistaken as nuclear tones, such as the one near the end of the statement at the 1.67 seconds time point in Figure 6 (which is indicated by an asterisk *). The effective change parameter, shown in Figure 7, instructs the model to ignore any F0 change in an interval that is less than the specified value. The default value is 20 Hz. Figure 8 shows an example. Despite the slight fall at the very end of the F0 contour, the model was able to correctly identify the low nuclear tone (L*).

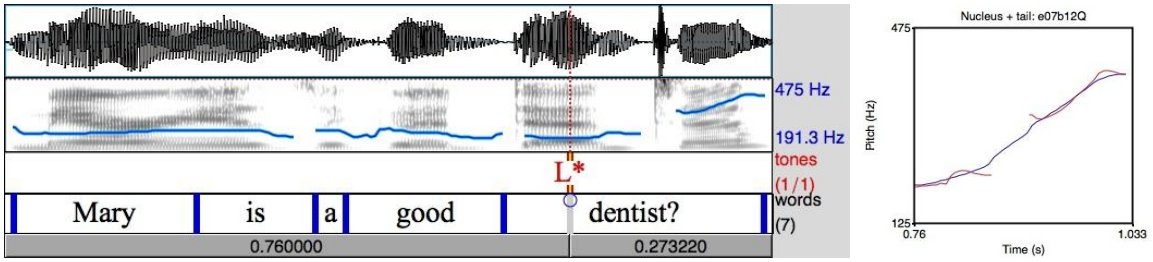

Figure 8: The F0 contour of 'Mary is a good dentist?' with the L* nuclear tone on the penultimate syllable [ˈdɛ̃n] (left), and the model's interpolated contour of the nucleus and tail (right)

2.2.3 Extraction

The extraction function extracts the auditory property measurements (i.e., timing, direction, and speed) from the sound tokens. By default, the pitch analysis and pitch range settings for the extraction function are the settings that were specified for the analysis function. If the user requests the model to extract one sound, it will display its auditory property measurements in the temporary output window so that they can be reviewed. If the user requests the model to apply the extraction function to all of the sound tokens, it will save the auditory property measurements to an output file that can be used by the training and testing functions in the next steps (or later on).

2.2.4 Training

Human listeners may give different degrees of attention to different auditory properties. For example, listeners may pay more attention to the direction of the nuclear tone (whether it is a fall or rise) if it serves as a better cue in identifying the sentence type than the relative timing of the nuclear tone in the intonational phrase. Therefore, direction might be weighted more than timing in distinguishing between statements and echo questions. To simulate this reality, the model assigns different weights to the auditory properties when calculating the auditory distance between a new token and an exemplar in memory.

The primary role of the training function is two-fold. 1) It enables the model to experience some of the tokens so that it will have some exemplars in memory for testing. 2) It trains the model to learn the weight distribution or the *generalized weights* of the auditory properties that would yield the best accuracy rate in categorizing new tokens. At the start of training, the model is given a set of *initial weights* of the auditory properties of the tokens. At the end of training, the

model not only has experienced the tokens that were presented to it but also has learned, to a degree, the general intonation patterns of these tokens as reflected in the shifting of the weights from the initial weights to the generalized weights. For example, in Figure 9, the window on the left shows the default initial weights of 50% for each of the auditory properties at the start of training. The window on the right shows the generalized weights of 30%, 40%, and 0% for speed, direction, and timing, respectively, at the end of training.

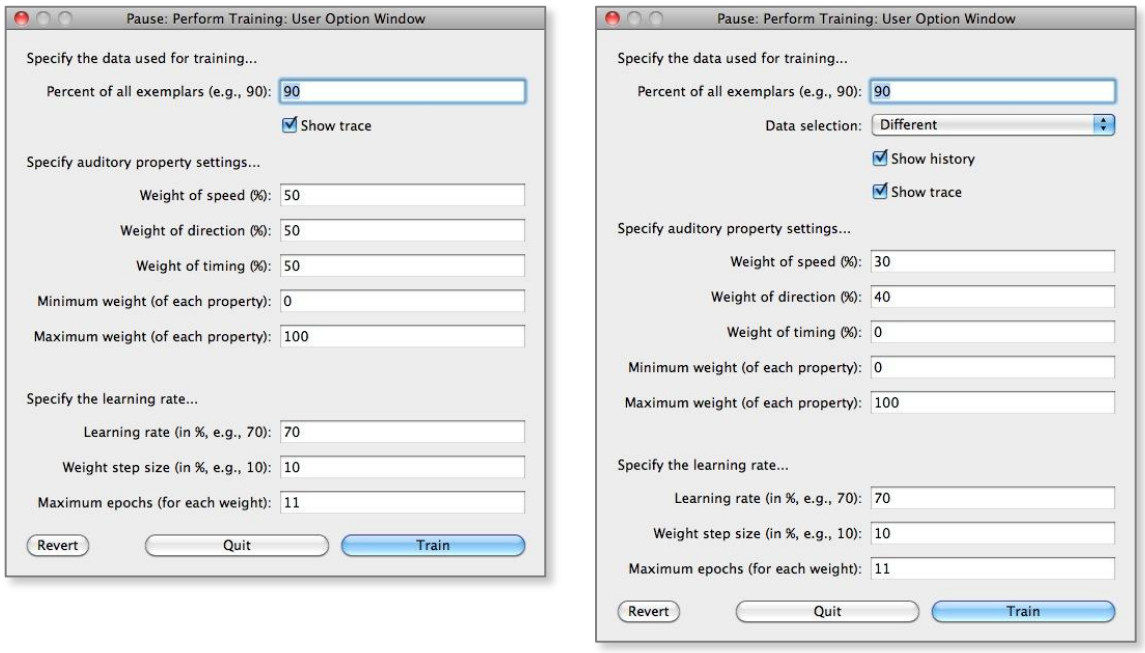

Figure 9: The parameter setting window for the training function before (left) and after (right) training

Using a trial-and-error approach, the model searches for an optimal set of generalized weights. From the initial set of weights, it increases or decreases the weight of one auditory property by one step size at a time, recomputing the similarity values with each step to determine if these values would yield a higher accuracy rate in categorizing the tokens than the previous step. It continues until it can no longer find a higher accuracy rate. The weight step size is how much the model increases or decreases the auditory property weight during each learning trial. The default step size is 10%. Larger step sizes help to speed up the search process with bigger jumps in weight, which is advantageous when the auditory property being considered is an ineffective cue. However, if the auditory property is an effective cue, larger step sizes could undesirably skip over an optimal weight.

In general, more training, learning, or experience tends to yield a higher accuracy rate in testing. By default, the number of tokens from the data samples that will be used for training is set to 90%. If the model learns to categorize the training tokens perfectly or too specifically, it might fail to recognize the general structure of the tokens. Therefore, the model is set, by default, to stop training once it reaches an accuracy rate of 70%. However, if the model fails to reach the default learning rate, training could continue forever. The maximum epochs parameter prevents this from happening by specifying the number of times that each auditory property's weight can be adjusted. The default setting is 11, which is the default maximum weight of an auditory property divided by the default weight step size and then added to one.

2.2.5 Testing

The testing function tests how accurately the model can categorize statements and questions from a set of sentences that are different from the training set. By default, it uses the generalized weights that the model has learned from training, as shown in Figure 10, in order to test how well the model generalizes to new tokens. However, if the user wants to find out whether another set of generalized weights would yield better performance, these weights can be adjusted and then the user can retest the model without retraining it. In this case, since the model did not learn the user-adjusted generalized weights, the test results are only meaningful in what-if analyses.

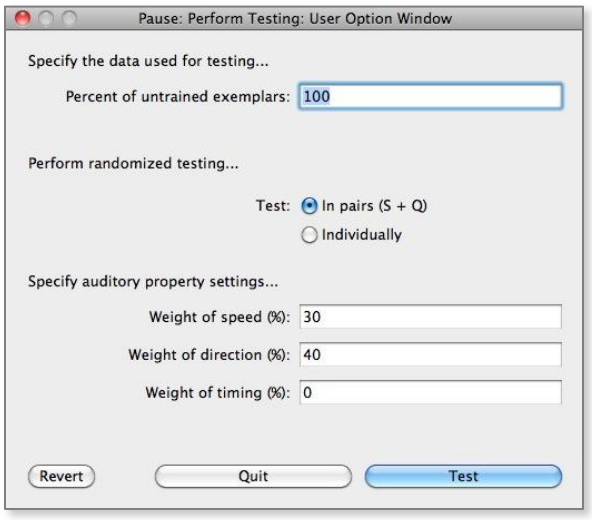

Figure 10: The parameter setting window for the testing function

2.2.6 Cross-validation

Cross-validation (Refaeilzadeh, Tang, & Liu 2009) is an approach to training and testing the model which tries to avoid *overfitting* the model such that it only recognizes the specifics of the data structure and fails to recognize the general structure. For example, some varieties of English express statements with a *high rising terminal* or *uptalk* intonation (Ladd 2008; Warren 2016). If none of the trained tokens that were presented to the model had the uptalk pattern, the model would generalize that all statements in English end with a falling intonation. Then, when the model encounters a token with uptalk in testing, it would fail to recognize that the token is a statement. Cross-validation attempts to avoid such over-specificity in the model through multiple training-and-test runs. In each run, a subset of the tokens serves as training data. The particular tokens used as training data rotate with each iteration of training in cross-validation, however, thus enabling the model to have eventually experienced all tokens as training tokens across the multiple runs. Similarly, cross-validation ensures that each token is tested once and only once across multiple runs. The average score across all runs is taken as a measure of how well the model has generalized to the tokens' intonation patterns.

3 Experiment

3.1 Goals

We trained and tested the model to find out 1) how well the model can learn to categorize statements and echo questions in English and 2) how sensitive the model is to the auditory properties (timing, direction, and speed) in distinguishing between statements and echo questions.

3.2 Methods

3.2.1 Speakers

Sixteen native speakers of Canadian English (8 male, 8 female, aged 18–23 years) produced the speech data that were used in this project. Participants were recruited from the University of Calgary's Introduction to Linguistics course and through flyers posted at the University of Calgary. They reported no visual, hearing, or speech impairments.

3.2.2 Materials

The speech data were recorded by the first author at the University of Calgary as part of the language corpus she developed for her graduate research there. This corpus includes 20 unique pairs of English statements and echo questions, produced by the 16 Canadian English speakers. Each paired statement and question were syntactically and lexically identical (e.g., 'Ann is a teacher. Ann is a teacher?'). Each speaker read the 20 pairs of sentences twice. The utterances were recorded in a sound-attenuated booth with high-quality recording equipment at a sampling rate of 44.1 kHz in a 16-bit mono channel and were saved to .wav files.

From this corpus, two speakers (one male, one female) were randomly selected from the subgroup of eight Canadian English speakers who were of 18 years of age. We chose younger speakers because they tend to use uptalk more (Sando 2009; Shokeir 2008), and felt that it would provide the model with a meaningful challenge to find out how well it could handle uptalk. These speakers originally produced 80 pairs of recorded sentences (2 speakers x 20 unique pairs of sentences x 2 readings). However, 16 of these pairs were excluded from the training and test data because Praat was unable to track the F0 values of some part of these sentences due to creaky voice.

3.2.3 Training

The initial weights (IWs) of the auditory properties can affect the model's success in finding an optimal set of generalized weights. Hypothetically, if the model begins training with 100% weight on timing and 0% weight on direction and speed, and it reaches the target learning rate before the other two auditory properties have an opportunity to gain some weight, we would not know if these two auditory properties matter in categorizing the sentences. Therefore, we trained the model with four different sets of initial weights, as listed in Table 1.

Additionally, how well the model generalizes depends largely on the training and test tokens. For example, if the training tokens have typical intonation patterns but the test tokens have atypical patterns, then the model would not generalize well and would likely perform poorly during testing. So, for each set of initial weights, the model was trained and tested 10 times. The training and test data were randomly selected each time. (In this initial experiment, we did not use cross-validation in order to find out how well the model would perform on each independent run.)

Table 1: Initial weights of the auditory properties for each set of IWs

	Initial Weights		
IW Set	Timing	Direction	Speed
T-100	100	0	0
D-100	0	100	0
S-100	0	0	100
TDS-50	50	50	50

For each run, 90% of the tokens were used for training. Training ceased when the accuracy rate in categorizing statements and questions (i.e., the learning rate) reached 70% or when the number of adjustments to an auditory property (i.e., maximum epochs) reached 11. The amount of adjustment to an auditory property during the weight distribution process was set to 10%.

3.2.4 Testing

For each of the 40 test runs (4 sets of initial weights x 10 runs), the model applied the set of generalized weights that was learned from training in the current run to the categorization of the remaining 10% of untrained tokens.

3.3 Results

3.3.1 Generalized Weights

Table 2 lists the mean, standard deviation (SD), and range – across all ten runs – of the generalized weights that the model came up with through training, with each set of initial weights. The strength of the direction cue is evident in the amount of weight shifted towards this cue through training. When the initial weight of direction was 100%, this weight remained at 100%. When the initial weight of direction was less than 100%, this weight increased. In contrast, speed does not appear to provide a strong cue to sentence type. When the initial weight of speed was 0%, this weight remained at 0%. When the initial weight of speed was greater than 0%, this weight decreased. On the other hand, timing appears to provide a stronger cue than speed but a weaker cue than direction. When the initial weight of timing was greater than 0%, this weight decreased. When the initial weight of timing was 0%, this weight increased only when the initial weight of direction was less than 100%; otherwise, this weight remained at 0%.

Table 2: Mean (SD, range) of the generalized weights resulting from training with each of the four sets of IWs

	Generalized Weights (%)		
IW Set	Timing	Direction	Speed
T-100	44 (33.7, 0–100)	11 (3.2, 10–20)	0 (0.0, 0–0)
D-100	0 (0.0, 0–0)	100 (0.0, 100–100)	0 (0.0, 0–0)
S-100	13 (13.4, 0–30)	79 (30.3, 0–100)	92 (13.2, 60–100)
TDS-50	19 (24.7, 0–80)	80 (21.1, 40–100)	36 (9.7, 20–50)

3.3.2 Categorization of Sentence Types

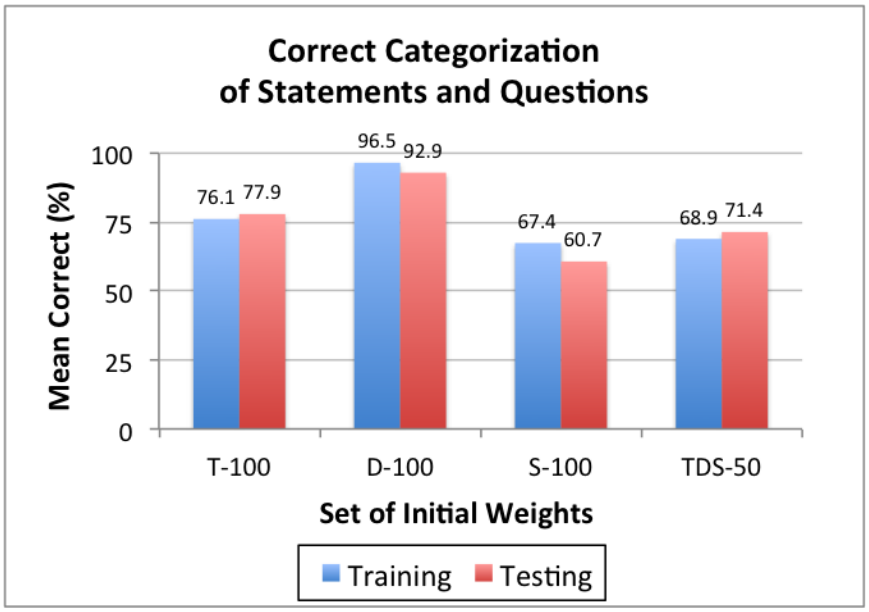

Figure 11: Correct categorization of statements and echo questions for each set of initial weights, averaged across ten runs

Figure 11 displays the mean percentage of statements and echo questions correctly categorized during training and testing over all ten runs. The differences between the training and test results are within 1.8–2.5%, indicating fairly consistent performances between training and testing. For all four sets of IWs, the model correctly categorized the two sentence types above 60%. It performed the best at 92.9% when IW was D-100. It performed the second best at 77.9% when IW was T-100. It performed the worst at 60.7% when IW was S-100. It performed slightly better at 71.4% when IW was TDS-50. (See Table 2 for the generalized weights resulting from using these IWs.) These results indicate that the model performed best when the most weight was placed on direction coupled with the least weight on speed. This finding is consistent with the preference for direction and the lack of preference for speed that were discussed in Section 3.3.1.

Table 3 lists the mean of correct categorization scores, along with the standard deviation (SD) and range for each set of initial weights. Testing reveals a wider range of scores than training, partly due to the 70% accuracy criterion placed on training. Also, there could be test tokens with unexpected patterns that the model could not handle well. The standard deviation is the smallest when IW was D-100, followed by TDS-50, then S-100, and finally T-100.

Table 3: Mean (SD, range) of correct categorization when trained with each of the four sets of IW

	Correct Categorization of Statements and Questions (%)					
IW Set	Training			Testing		
T-100	76.1	(10.6,	70.1–96.5)	77.9	(12.3,	64.3–100.0)
D-100	96.5	(0.6,	95.6–97.4)	92.9	(4.8,	85.7–100.0)
S-100	67.4	(2.7,	64.0–70.2)	60.7	(12.7,	35.7–78.6)
TDS-50	68.9	(2.2,	64.9–72.8)	71.4	(8.9,	57.1–85.7)

With the initial weights set at T-100 or D-100, all ten runs reached the target learning rate of 70% during training. However, with the initial weights set at S-100, seven out of the ten runs did not reach this threshold in training, and only three runs did (64.0–69.3% vs. 70.2%). Interestingly, the seven runs that failed to reach the training performance criterion performed better than those remaining three runs in testing (64.3–78.6% vs. 35.7–50%). Similarly, with the initial weights set at TDS-50, six out of the ten runs did not reach the target learning rate of 70% during training, while only four runs did (64.9–69.3% vs. 70.2–72.8%). As above, the six runs that failed to reach the training criterion performed slightly better than the other four runs in testing (64.3–85.7% vs. 57.1–78.6%). This means that when the initial weight of speed was equal to or greater than both timing and direction, it took longer (or requires more epochs) for the model to derive a best-fitting set of weights for the training data.

4 Conclusion

This research project investigated whether an exemplar-based model could learn to categorize statements and echo questions in English, based on intonation alone. We created a computational model that learned to categorize English statements and echo questions by comparing the similarity of new tokens to previously classified tokens that were stored as exemplars in memory. To determine the auditory similarity between two sentences, the model used three properties: 1) the relative timing of the nuclear tone in the sentence, 2) the direction of the post-nuclear intonation contour, and 3) the speed at which the post-nuclear intonation contour rises or falls. The model was presented with 64 pairs of sentences that were produced by one male and one female speaker of Canadian English, but the F0 contours of these sentences were not normalized for each speaker prior to training and testing. The highest-performing version of the model correctly categorized up to 92% of the unheard sentences presented in testing, averaged across ten training-and-test runs. This preliminary experimental result demonstrates that it is feasible to develop an exemplar-based computational model that can learn to categorize statements and echo questions in English, without normalizing F0 to account for speaker variability.

However, much of the model's performance in testing depended on the specific properties of the sentences that were tested and the generalized weights that were assigned to those properties. The model performed the best when it focused all of its attention on the direction auditory property. It is interesting to think that this exemplar-based model – which does not *filter out* any information in its representation of speech exemplars in memory – can develop behaviour where it ignores potentially useful acoustic cues and focuses on only one, crucial property of the signal in perception.

We see a number of fruitful opportunities for future research that can expand upon this initial, promising result. First of all, in order to determine whether exemplar theory could provide a realistic account for the human perception of intonation, it would be necessary to conduct a similar identification task of statements and echo questions on human listeners and then compare the human performance with the model's performance. Secondly, it would be practical to include more speakers in order to increase the variability of the test sentences for the model. Thirdly, the model should also include other auditory properties or more detailed acoustic information in its similarity calculations, such as F0 height, to see if they would affect the model's performance. Lastly, it would be insightful to explicitly test this model on uptalk intonation to see how well it can generalize between statements with and without uptalk, and also distinguish between the similar intonation patterns of uptalk statements and echo questions.

References

Boersma, P., & Weenink, D. (2013, May 30). *Praat: doing phonetics by computer*. [Computer application, version 5.3.51]. Retrieved from http://www.praat.org

Chow, U. Y., & Winters, S. J. (2015). Exemplar-based classification of statements and questions in Cantonese. In The Scottish Consortium for ICPhS 2015 (Ed.), *Proceedings of the 18th International Congress of Phonetic Sciences*. Glasgow, UK: the University of Glasgow. ISBN 978-0-85261-941-4. Paper number 0987.

Fant, G. (1972). Vocal tract wall effects, losses, and resonance bandwidths. *Speech Transmission Laboratory Quarterly Progress and Status Report, 2*(3), 28–52.

Goldinger, S. D. (1998). Echoes of echoes? An episodic theory of lexical access. *Psychological Review, 105*(2), 251–279.

Johnson, K. (1997). Speech perception without speaker normalization: An exemplar model. In K. Johnson, & J. W. Mullennix (Eds.), *Talker variability in speech processing* (pp. 145–165). San Diego: Academic Press.

Ladd, D. R. (2008). *Intonational phonology (2nd ed.)*. Cambridge: Cambridge University Press.

Nosofsky, R. M. (1988). Exemplar-based accounts of relations between classification, recognition, and typicality. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 14*, 700–708.

Peterson, G., & Barney, H. (1952). Control methods used in a study of the vowels. *Journal of the Acoustical Society of America, 24*(2), 175–184.

Pierrehumbert, J., & Hirschberg, J. (1990). The meaning of intonation contours in the interpretation of discourse. In P. R. Cohen, J. Morgan, & M. Pollack (Eds.), *Plans and intentions in communication and discourse* (pp. 271–311). Cambridge: MIT Press.

Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation. In L. Liu, & M. T. Zsu (Eds.), *Encyclopedia of database systems* (pp. 532–538). Springer Publishing Company Incorporated.

Sando, Y. T. (2009). Upspeak across Canadian English accents: Acoustic and sociophonetic evidence. *Proceedings of the 2009 annual conference of the Canadian Linguistic Association*.

Shokeir, V. (2008). Evidence for the stable use of uptalk in South Ontario English. *University of Pennsylvania Working Papers in Linguistics, 14(2/4)*.

Simpson, A. P. (2009). Phonetic differences between male and female speech. *Language and Linguistics Compass, 3(2)*, 621–640.

Walsh, M., Schweitzer, K., & Schauffer, N. (2013). Exemplar-based pitch accent categorisation using the Generalized Context Model. *Proceedings of the 14th Annual Conference of the International Speech Communication Association*, 258–262.

Warren, P. (2016). *Uptalk: The phenomenon of rising intonation*. Cambridge: Cambridge University Press.

Wells, J. C. (2006). *English intonation: An introduction*. Cambridge: Cambridge University Press.