PROJECT MUSE®

Individual variation in the structure of bilingual grammars: Supplementary material

Clara Cohen, Syed Waqar Nabi, Catherine F. Higham, Michael Putnam,
Gerrit Jan Kootstra, Janet G. van Hell

LANGUAGE	
A JOURNAL OF THE LINGUISTIC SOCIETY OF AMERICA	
VOLUME 97, NUMBER 4, DECEMBER 2021	
Individual variation in the structure of bilingual grammars: Supplementary material	107
Language and cognitive control in bilinguals	108
Language and cognitive control in bilinguals: A review	109
Language and cognitive control in bilinguals: A review	110
Language and cognitive control in bilinguals: A review	111
Language and cognitive control in bilinguals: A review	112
Language and cognitive control in bilinguals: A review	113
Language and cognitive control in bilinguals: A review	114
Language and cognitive control in bilinguals: A review	115
Language and cognitive control in bilinguals: A review	116
Language and cognitive control in bilinguals: A review	117
Language and cognitive control in bilinguals: A review	118
Language and cognitive control in bilinguals: A review	119
Language and cognitive control in bilinguals: A review	120
Language and cognitive control in bilinguals: A review	121
Language and cognitive control in bilinguals: A review	122
Language and cognitive control in bilinguals: A review	123
Language and cognitive control in bilinguals: A review	124
Language and cognitive control in bilinguals: A review	125
Language and cognitive control in bilinguals: A review	126
Language and cognitive control in bilinguals: A review	127
Language and cognitive control in bilinguals: A review	128
Language and cognitive control in bilinguals: A review	129
Language and cognitive control in bilinguals: A review	130
Language and cognitive control in bilinguals: A review	131
Language and cognitive control in bilinguals: A review	132
Language and cognitive control in bilinguals: A review	133
Language and cognitive control in bilinguals: A review	134
Language and cognitive control in bilinguals: A review	135
Language and cognitive control in bilinguals: A review	136
Language and cognitive control in bilinguals: A review	137
Language and cognitive control in bilinguals: A review	138
Language and cognitive control in bilinguals: A review	139
Language and cognitive control in bilinguals: A review	140
Language and cognitive control in bilinguals: A review	141
Language and cognitive control in bilinguals: A review	142
Language and cognitive control in bilinguals: A review	143
Language and cognitive control in bilinguals: A review	144
Language and cognitive control in bilinguals: A review	145
Language and cognitive control in bilinguals: A review	146
Language and cognitive control in bilinguals: A review	147
Language and cognitive control in bilinguals: A review	148
Language and cognitive control in bilinguals: A review	149
Language and cognitive control in bilinguals: A review	150
Language and cognitive control in bilinguals: A review	151
Language and cognitive control in bilinguals: A review	152
Language and cognitive control in bilinguals: A review	153
Language and cognitive control in bilinguals: A review	154
Language and cognitive control in bilinguals: A review	155
Language and cognitive control in bilinguals: A review	156
Language and cognitive control in bilinguals: A review	157
Language and cognitive control in bilinguals: A review	158
Language and cognitive control in bilinguals: A review	159
Language and cognitive control in bilinguals: A review	160
Language and cognitive control in bilinguals: A review	161
Language and cognitive control in bilinguals: A review	162
Language and cognitive control in bilinguals: A review	163
Language and cognitive control in bilinguals: A review	164
Language and cognitive control in bilinguals: A review	165
Language and cognitive control in bilinguals: A review	166
Language and cognitive control in bilinguals: A review	167
Language and cognitive control in bilinguals: A review	168
Language and cognitive control in bilinguals: A review	169
Language and cognitive control in bilinguals: A review	170
Language and cognitive control in bilinguals: A review	171
Language and cognitive control in bilinguals: A review	172
Language and cognitive control in bilinguals: A review	173
Language and cognitive control in bilinguals: A review	174
Language and cognitive control in bilinguals: A review	175
Language and cognitive control in bilinguals: A review	176
Language and cognitive control in bilinguals: A review	177
Language and cognitive control in bilinguals: A review	178
Language and cognitive control in bilinguals: A review	179
Language and cognitive control in bilinguals: A review	180
Language and cognitive control in bilinguals: A review	181
Language and cognitive control in bilinguals: A review	182
Language and cognitive control in bilinguals: A review	183
Language and cognitive control in bilinguals: A review	184
Language and cognitive control in bilinguals: A review	185
Language and cognitive control in bilinguals: A review	186
Language and cognitive control in bilinguals: A review	187
Language and cognitive control in bilinguals: A review	188
Language and cognitive control in bilinguals: A review	189
Language and cognitive control in bilinguals: A review	190
Language and cognitive control in bilinguals: A review	191
Language and cognitive control in bilinguals: A review	192
Language and cognitive control in bilinguals: A review	193
Language and cognitive control in bilinguals: A review	194
Language and cognitive control in bilinguals: A review	195
Language and cognitive control in bilinguals: A review	196
Language and cognitive control in bilinguals: A review	197
Language and cognitive control in bilinguals: A review	198
Language and cognitive control in bilinguals: A review	199
Language and cognitive control in bilinguals: A review	200

Language, Volume 97, Number 4, December 2021, (Article)

Published by Linguistic Society of America

DOI: <https://doi.org/10.1353/lan.2021.0077>

➔ *For additional information about this article*

<https://muse.jhu.edu/article/842381/summary>

Individual variation in the structure of bilingual grammars:

Supplemental materials

Clara Cohen Syed Waqar Nabi Catherine F. Higham

Michael Putnam Gerrit Jan Kootstra Janet van Hell

June 18, 2021

A How to calculate harmony

A.1 In a monolingual grammar

The calculation of harmony in a GSC framework minimally involves one set of parameters: the weights assigned to each constraint. To calculate the harmony of each potential output, the violations of each constraint associated with each output are multiplied by their respective constraint’s weight, and the negative sum of the weighted violations is the harmony. Thus, in the toy grammar in Table 1, the highest-weighted constraint is MAX, which means that the middling-sized 2.5 violations—substantially lower than the maximum of 100, but higher than the violations of two other constraints—have the most extreme contribution to harmony of 23.75 units. The 0-weighted constraint MATCHSWITCH, by contrast, can have 100 violations without affecting the harmony score at all, because the violations are wiped out when they are multiplied by 0.

The harmony resulting from the output of Y in response to input X in this toy grammar is -42.15. This is calculated by taking the negative sum of each weighted constraint. Thus, the harmony is $-(0.4 + 11.7 + 6 + 0.3 + 23.75 + 0) = -42.15$.

This grammar would be sufficient if we were modeling a monolingual speaker, who presumably only has one set of constraints with one set of weights¹. In a bilingual speaker, however, we must

¹ This presumption is of course an oversimplification: Even in a monolingual speaker, it is likely that the different speech registers associated with different situational contexts arise from the fact that the speaker’s constraint weights can be pushed around by external factors, even if in a more constrained fashion.

Table 1: Calculations of harmony for hypothetical input-output pair, with violations weighted by sample weights.

Input=X	SPECLEFT 0.4	HEADRIGHT 1.3	HEADLEFT 3.0	MATCHPROMPT 0.1	MAX 9.5	MATCHSWITCH 0	Harmony
Output=Y	1	9	2	3	2.5	100	
	$1 \times 0.4 = 0.4$	$9 \times 1.3 = 11.7$	$2 \times 3.0 = 6$	$3 \times 0.1 = 0.3$	$2.5 \times 9.5 = 23.75$	$100 \times 0 = 0$	-42.15

Table 2: Sample calculation for the harmony contributions produced by HEADLEFT in an unbalanced, L1-dominant bilingual speaker producing an SVO utterance in L1.

Input = SVO	HEADLEFT			
Language	Activation	Weight	Activation-adjusted weight	
L1	.75	10	$.75 \times 10 = 7.5$	
L2	.25	5	$.25 \times 5 = 1.25$	
General	$7.25 + 1.25 = 8.75$			
Output = SVO	L1 violations	L2 violations	General violations	All violations
	1	0	$1+0 = 1$	
	$1 \times 7.5 = 7.5$	$0 \times 1.25 = 0$	$1 \times 8.75 = 8.75$	$7.5 + 0 + 8.75 =$ 16.25

allow for the fact that different languages have different grammars, and so the constraints and weights that make up both grammars must be represented, as proposed in Goldrick et al. (2016).

A.2 In a bilingual grammar

To understand how both grammars interact in this model, let us first consider a toy example², summarized in Table 2, illustrating how the weighted violations of a single constraint are calculated. In this example, the speaker is not a balanced bilingual, and so different activation values are assigned to each of the languages. Here we use .75 for L1, the speaker’s dominant language, and .25 for L2. Since the two languages have different grammars, their weights for the constraint in question—here, HEADLEFT—are also different. We assume for this example that L1 has a weight of 10 for HEADLEFT, while L2 has a weight of 5. This corresponds to a system when L1 prefers verbs to be at the left edge of the clause more strongly than L2. The two language-specific weights, 10 for L1 and 5 for L2, are then adjusted by their respective language’s activation value. Thus, the activation-adjusted weight of for L1 is 7.5 ($= .75 \times 10$), and the activation-adjusted weight for L2 is 1.25 ($= .25 \times 5$). These activation-adjusted weights are then summed to produce a language-general weight of 8.75 ($= 7.5 + 1.25$).

The bottom half of Table 2 shows how these language-specific and language-general weights combine with the violations incurred by a particular utterance. Suppose a speaker produces an SVO utterance in L1. This will generate 1 violation for the L1-specific HEADLEFT, because the utterance contains an L1 head one position removed from the left edge of the clause. No violations will be incurred for L2-specific HEADLEFT, because the head was not uttered in L2. These two language-specific violation values—1 and 0—are then summed, to produce a language-general HEADLEFT violation count of 1.

Next, each of the three violation values (1 for L1, 0 for L2, 1 for language-general) is multiplied by the appropriate activation-adjusted weight. Thus, the 1 violation for L1 is multiplied by 7.5, the 0 violations for L2 are multiplied by 1.25, and the 1 language-general violation is multiplied by 8.75. These weighted violations are then summed, producing a contribution to the final harmony

²adapted from Goldrick et al. (2016, pg. 16).

score of 16.25. This process is repeated for all constraints, and the candidate’s final harmony score is again the negative sum of the weighted violations of each constraint, as calculated above in §A.1.

A.3 Weighting MAX violations with ϵ and δ

The calculation of MAX is slightly different from the other constraints. In our implementation, we assume that all constituents to be produced—subject, direct object, and verb—are activated in both English and Dutch. This means that, for a transitive utterance, six constituents are activated, but only three are produced. How should the three violations of MAX, incurred by the three omitted constituents, be calculated? We decided to use MAX to capture the dynamic nature of the language context across different prompt-languages and different switch-conditions, by including another set of language engagement parameters, ϵ and δ .

Consider the following situation. The participant hears a confederate describe a picture of a girl chasing a donkey, as in (1) below:

- (1) Een grappig plaatje, waarop het meisje *the donkey chases*
A funny picture, on which the girl *the donkey chases*
A funny picture on which the girl chases the donkey.

In this utterance, the confederate used an SOV word order to describe the picture, and switched to English after the first constituent. The utterance can therefore be described as $S_D O_E V_E$. The participant is then prompted with the same preamble (*Een grappig plaatje, waarop*) to describe a picture of a boy calling a lion, using at least one word of English in their response. Both the confederate’s word order on the previous trial and the preamble of the sentence encourage an SOV description. The prompt to use at least one word of English encourages a code-switch, and the confederate’s switch location on the previous trial primes the participant to produce a code-switch after the first constituent in their own response. In our implementation, we capture these factors by coding the input to the utterance as $S_D O_E V_E$, indicating an SOV word order, with a switch location after the first constituent. Since this input contains one Dutch constituent and two English constituents, the ϵ and δ language engagement parameters for this utterance are 0.55 and 0.45, respectively (see Table 1 in the main article text). This weighting means that omitting English constituents will violate MAX more severely than omitting Dutch constituents.

Table 3 shows how the calculation proceeds. As in the previous example of the calculation of HEADLEFT, we assume the same L1 (Dutch) and L2 (English) activations of .75 and .25, representing a Dutch-dominant bilingual. We also assume, arbitrarily for this example, that the language-specific weights for MAX are 5 for Dutch and 9 for English, reflecting an English-dominant bilingual. The activation-adjusted weights are calculated in the same way as in Table 2, so there is nothing new about the first row in Table 3. The bottom three rows, however, show how violations

Table 3: Sample calculation for the harmony contributions produced by MAX when a response is primed with a switch after the first constituent.

Input= $S_D O_E V_E$ $\epsilon = .55$; $\delta = .45$		MAX		
Language	Activation	Weight	Activation-adjusted weight	
Dutch	.75	5	$.75 \times 5 = 3.75$	
English	.25	9	$.25 \times 9 = 2.25$	
General			$3.75 + 2.25 = 6$	
1) $S_D O_E V_E$	Dutch $2 \times \delta = .9$ $.9 \times 3.75 = 3.375$	English $1 \times \epsilon = .55$ $.55 \times 2.25 = 1.238$	General $.9 + .55 = 1.45$ $1.45 \times 6 = 8.7$	All $3.375 + 1.238 + 8.7 =$ 13.313
2) $V_E S_E O_D$	Dutch $2 \times \delta = .9$ $.9 \times 3.75 = 3.375$	English $1 \times \epsilon = .55$ $.55 \times 2.25 = 1.238$	General $.9 + .55 = 1.45$ $1.45 \times 6 = 8.7$	All $3.375 + 1.238 + 8.7 =$ 13.313
3) $V_D S_E O_D$	Dutch $1 \times \delta = .45$ $.45 \times 3.75 = 1.688$	English $2 \times \epsilon = 1.10$ $1.1 \times 2.25 = 2.475$	General $.45 + 1.1 = 1.55$ $1.55 \times 6 = 9.3$	All $1.688 + 2.475 + 9.3 =$ 13.463

of MAX are further adjusted by the ϵ and δ values.³ In candidate 1, the participant switched at the primed location, producing an SOV utterance with a Dutch S and an English O and V. Since O and V are both English, the speaker omitted two Dutch constituents, the Dutch O and the V, and therefore incurs a Dutch-specific MAX violation of $2 \times \delta = 0.9$. Since S was Dutch, the speaker omitted the English S, and therefore incurs an English-specific MAX violation of $1 \times \epsilon = 0.55$. These violations are then summed to arrive at the language-general MAX violation of 1.45. All three of these violations are then weighted by their respective activation-adjusted MAX weights. For Dutch, this results in a weighted violation of $.9 \times 3.75 = 3.375$; For English, $.55 \times 2.25 = 1.238$; and for the language-general constraint, $1.45 \times .6 = 8.7$. These three adjusted violations are then summed to produce a final harmony contribution of 13.313.

In candidate 2, the speaker switched immediately into English, and then switched back into Dutch for the last constituent. Since this utterance has exactly the same number of English and Dutch constituents as candidate 1, the final harmony calculation is identical. In other words, we assume that the mechanism of switch-location priming depends not on switching at the same location as primed, but rather on producing the same number of constituents in the target language as would be produced by switching at that location. This means that switching between languages is governed by the relative activation of the lexical items in a given utterance (ϵ and δ), rather than a specific reference to the actual process of switching. (Note that the word order is VSO, not SOV, but that violation will be handled by the MATCHPROMPT constraint, not by MAX).

Finally, in candidate 3, the participant produced an utterance with VSO word order that switched at the primed location into English, but then switched back again into Dutch. The result is an utterance containing two Dutch constituents (omitting one) and only one English constituent (omitting two). Since ϵ is greater than δ , the unweighted penalty is higher for omitting an

³These examples were chosen to illustrate the behavior of this model structure, and are not representative of the actual utterances speakers commonly produced for this input.

Table 4: Raw weights and proficiency-adjusted weights for the six constraints

Raw	SPECLEFT	HEADRIGHT	HEADLEFT	MATCHPROMPT	MAX	MATCHSWITCH
En	5.4	9.0	12.1	-2.0	4.2	-0.1
Du	3.2	6.0	7.4	5.0	3.9	0.8
Adjusted						
En ($\times .4$)	2.16	3.60	4.84	-0.80	1.68	-0.04
Du ($\times .6$)	1.92	3.60	4.44	3.00	2.34	0.48
General	4.08	7.20	9.28	2.20	4.02	0.44

English constituent than omitting a Dutch constituent: 1.10 in the English column of the third row, compared to .45 in the Dutch column. If the language activations and the constraint weights were roughly equivalent, this would lead to a higher final harmony score for candidate 3 than candidates 1 and 2. However, the activation for Dutch, 0.75, is much higher than the activation for English, 0.25, so the higher English penalty is down-weighted relative to the lower Dutch penalty. As a result, the activation-adjusted weights for the two languages end up being more similar in candidate 3 than in candidates 2 and 1. With the additional contribution of the language-general weight further smoothing out the difference between English-specific and Dutch-specific violations, the result is that candidate 3 and candidates 1 and 2 end up having quite similar final harmony scores. In this way, the model allows us to capture a tendency for speakers to prefer their dominant language, sometimes switching back into it even after they switch away in the prompted position.

A.4 Calculating a full harmony score for two candidates.

We can now put the pieces together and walk through a full harmony calculation for two sample candidates. To start, we will need a set of twelve weights—both English-specific and Dutch-specific for each of the six different constraints—and some English and Dutch proficiency values to adjust the weights. For simplicity’s sake, we will use the values, rounded to the nearest tenth, from Table 2 in the main text. These are given in the top half of Table 4.

Recall, however, that the proficiency of the speaker is not equivalent in each language. For this example, we will assume that the English proficiency is .4 and the Dutch proficiency is .6. Accordingly, the English-specific and Dutch-specific weights must be multiplied by the respective language proficiency, and then added together to produce the language-general weight. The result of these adjustments are given in the bottom half of Table 4.

To illustrate the calculation of harmony, we must have some sample candidates to go with an input and an output. Again for the sake of simplicity, we will use the input candidate from Table 2 in the main text—*en-sov-1*. In other words, it will be a prompt with an English preamble that, were it in Dutch, would prompt SOV word order, with a code-switch primed to take place after the first constituent. The two output candidates will be, first, candidate 1 from Table 2 ($|S_D O_D V_D$), and second, a distinctly different candidate: $V_E |S_D| O_E$. In other words, candidate 1 will be an

utterance with SOV word order and an immediate code-switch into Dutch, earlier than the primed location; while candidate 2 will be an utterance with VSO word order and two switches, one after the V—in the primed location—and a second after the S.

Table 5 walks through each constraint, in each language, for each candidate, and assigns raw violations, with reasoning provided for each assignment.

Table 5: Logic of violation assignments

Constraint	Language	Candidate	Violations	Reason
SPECLEFT	English	$ S_D O_D V_D$	0	S is not in English
SPECLEFT	English	$V_E S_D O_E$	0	S is not in English
SPECLEFT	Dutch	$ S_D O_D V_D$	0	S is at left edge of clause
SPECLEFT	Dutch	$V_E S_D O_E$	1	S is one position from left edge
HEADRIGHT	English	$ S_D O_D V_D$	0	V is not in English
HEADRIGHT	English	$V_E S_D O_E$	2	V is two positions from right edge of clause
HEADRIGHT	Dutch	$ S_D O_D V_D$	0	V is at right edge of clause
HEADRIGHT	Dutch	$V_E S_D O_E$	0	V is not in Dutch
HEADLEFT	English	$ S_D O_D V_D$	0	V is not in English
HEADLEFT	English	$V_E S_D O_E$	0	V is at left edge of clause
HEADLEFT	Dutch	$ S_D O_D V_D$	2	V is two positions from left edge of clause
HEADLEFT	Dutch	$V_E S_D O_E$	0	V is not in Dutch
MATCHPROMPT	English	$ S_D O_D V_D$	0	Candidate fully matches prompted order
MATCHPROMPT	English	$V_E S_D O_E$	1.33	VSO order requires 2 position swaps to match prompted SOV order; $2/3$ of utterance is English; $2 \times 2/3 = 4/3 = 1.33$
MATCHPROMPT	Dutch	$ S_D O_D V_D$	0	Candidate fully matches prompted order
MATCHPROMPT	Dutch	$V_E S_D O_E$	0.67	VSO order requires 2 position swaps to match prompted SOV order; $1/3$ of utterance is Dutch; $2 \times 1/3 = 2/3 = .67$
<i>The primed code-switch location would contain one English constituent following the English preamble and before the switch; and two Dutch constituents, following the switch. This corresponds to an ϵ value of .45, and a δ value of .55 (see Table 1 of the main text for more details).</i>				
MAX	English	$ S_D O_D V_D$	1.35	The all-Dutch response means that three English constituents have been lost. $3 \times .45 = 1.35$
MAX	English	$V_E S_D O_E$	0.45	The response contains two English constituents out of a possible three. This means that one English constituent has been lost. $1 \times .45 = .45$
MAX	Dutch	$ S_D O_D V_D$	0	The all-Dutch response means that no Dutch constituents have been lost. $0 \times .55 = 0$
MAX	Dutch	$V_E S_D O_E$	1.1	The response contains one Dutch constituent out of a possible three. This means that two Dutch constituents have been lost. $2 \times .55 = 1.1$
MATCHSWITCH	English	$ S_D O_D V_D$	0	response does not omit a prompted switch

Input: en-sov-1	SPECLEFT	HEADRIGHT	HEADLEFT	MATCHPROMPT	MAX	MATCHSWITCH	Harmony
Adjusted En ($\times 4$)	2.16	3.60	4.84	-0.80	1.68	-0.04	
Adjusted Du ($\times 6$)	1.92	3.60	4.44	3.00	2.34	0.48	
General	4.08	7.20	9.28	2.20	4.02	0.44	
1. $ S_D O_D V_D$	0×2.16 $+0 \times 1.92$ $+0 \times 4.08$ $=0$	0×3.6 $+0 \times 3.6$ $+0 \times 7.20$ $=0$	0×4.84 $+2 \times 4.44$ $+2 \times 9.28$ $=27.44$	$0 \times -.8$ $+0 \times 3.0$ $+0 \times 2.2$ $=0$	1.35×1.68 $+0 \times 2.34$ $+1.35 \times 4.02$ $=7.70$	0×-0.04 $+0 \times 0.48$ $+0 \times 0.44$ $=0$	$-(27.44$ $+ 7.70)$ $=-35.14$
2. $V_E S_D O_E$	0×2.16 $+1 \times 1.92$ $+1 \times 4.08$ $=6$	2×3.6 $+0 \times 3.6$ $+2 \times 7.20$ $=21.6$	0×4.84 $+0 \times 4.44$ $+0 \times 9.28$ $=0$	$1.33 \times -.8$ $+0.67 \times 3.0$ $+2 \times 2.2$ $=5.35$	0.45×1.68 $+1.1 \times 2.34$ $+1.55 \times 4.02$ $=9.56$	0×-0.04 $+0 \times 0.48$ $+0 \times 0.44$ $=0$	$-(6$ $+21.6$ $+5.35$ $+9.56)$ $=-42.51$

Table 6: Multiplying the raw violations by the proficiency-adjusted weights, and taking the negative weighted sum to calculate harmony for each candidate.

MATCHSWITCH	English	$V_E S_D O_E$	0	response does not omit a prompted switch
MATCHSWITCH	Dutch	$ S_D O_D V_D$	0	response does not omit a prompted switch
MATCHSWITCH	Dutch	$V_E S_D O_E$	0	response does not omit a prompted switch

For each constraint, we now multiply each language-specific violation by its adjusted language-specific weight (from the bottom half of Table 4), and then add those violations together and multiply the sum by the language-general weight. These three values—the two language specific weighted violations and the one language-general weighted violation—are then added together, as illustrated in Table 2, to produce each constraint’s contribution to the final harmony score. This full process is illustrated in Table 6. Note that the weights on the top of the table represent the *adjusted* weights, reflecting the contribution of unequal proficiency, rather than the raw weights produced by the optimization algorithm (see §B).

A.5 Transforming Harmony into probability

After the harmony is calculated for each candidate, the harmony values of all candidates $H(c)$ are transformed into a predicted usage probability. According to Note 4, Goldrick et al. (2016, pg. 864). the formula for converting the harmony value of a candidate, $H(c)$, is

$$\frac{e^{\frac{H(c)}{T}}}{\sum_x e^{\frac{H(x)}{T}}} \quad (1)$$

Where T is a parameter that can take a number of different weights. For our purposes, we followed Goldrick et al. (2016) in assigning to T the value of 1, which makes the formula somewhat simpler:

$$\frac{e^{H(c)}}{\sum_x e^{H(x)}} \quad (2)$$

In other words, we exponentiated each harmony value, and then divided it by the sum of all candidates’ exponentiated harmony values. Since the total harmony value is the *negative* sum of the weighted violations for each constraint, more violations will correspond to lower—i.e., more negative—harmony values. Then, the larger—i.e., the less negative—a given harmony value is, the larger its exponentiated value will be, and hence the greater its predicted probability. In this way candidates with more violations—or violations of constraints with greater weights—are assigned lower usage probabilities.

In this toy example, assuming that the two candidates in Table 6 were the only two possible utterances, we would calculate the probability distribution as follows.

First, we exponentiate the two harmony values. For Candidate 1, this gives $e^{-35.14} = 5.4 \times 10^{-16}$. For Candidate 2, this gives $e^{-42.51} = 3.5 \times 10^{-19}$. Next, we add together these two exponentiated values: $5.4 \times 10^{-16} + 3.5 \times 10^{-19} = 5.48 \times 10^{-16}$. Finally, we divide by this sum the exponentiated harmony value, to arrive at a predicted probability for each candidate. Note that, since the exponentiated values differ by three orders of magnitude, we can expect a similarly large difference in usage probability once they are divided by the summed denominator. And, indeed, this is what we find: For Candidate 1 the predicted probability is $\frac{5.4 \times 10^{-16}}{5.48 \times 10^{-16}} = .9994$, while for Candidate 2 this is $\frac{3.5 \times 10^{-19}}{5.48 \times 10^{-16}} = .0006$.

Since Candidate 2 employed a word order that is generally constrained, not consistent with the syntax of the prompt, and switched multiple times in a single utterance, it is entirely reasonable that its usage probability would be so small—especially when compared with an alternative candidate that departs from the prompted utterance only by changing its switch location to be slightly closer to the preamble. As other, more plausible alternatives are included in this calculation, the denominator of the probability transformation will increase because the inclusion of more candidates will add more terms to the sum. This means that all predicted probabilities will go down: The same probability mass will need to be distributed over more candidates.

The predicted probability distribution across all candidates is then used to evaluate the model. The more closely this probability distribution approximates the actual proportions of usages produced by the participants across all utterances, the better the fit of the model.

B The optimization algorithm

The optimization algorithm was conducted in the R programming environment (version 3.5.0, [R Core Team, 2019](#)). To start the process, each constraint weight was set at an initial value randomly

Raw	Old SPECLEFT	Jump	New SPECLEFT
En	5.4	0.57	5.97
Du	3.2	NA	3.2
Adjusted			
En ($\times .4$)	2.16		2.39
Du ($\times .6$)	1.92	NA	1.92 (no change)
General	4.08		4.31

Table 7: Updated weight calculations from Table 4 after a jump of 0.57 in the optimization procedure. General weights are calculated by simply adding together the adjusted language-specific weights.

selected from the uniform distribution between 0 and 10. The algorithm then cycled through each of the 12 language specific weights, and randomly selected a jump from the uniform distribution ranging from -2 to 2. Each jump was therefore equally likely to be positive or negative in value, and equally likely to be any size from -1 to 1. The current constraint weight was then changed by adding that jump to its value. From this slightly adjusted set of language-specific constraints, language-general constraint weights were re-calculated and from these new weights the harmony values for each of the 1728 prompt-response pairs were re-calculated, allowing the subsequent re-calculation of the predicted probability distribution across responses.

B.1 Example recalculation of probability distribution

As an example, consider the raw weights from Table 4, which provide the core of the harmony calculations illustrated in Table 6. The first jump in our optimization algorithm would target the English-specific SPECLEFT weight, choosing some new value from the uniform distribution ranging from -1 to 1. Suppose the jump appears as 0.57. We therefore add this value to the raw weight for the English SPECLEFT, changing it from its original value of 5.4 (top half of Table 4) to a new weight of 5.97 ($5.4 + 0.57 = 5.97$). This results in the new set of raw and adjusted weights for English-specific SPECLEFT, as laid out in Table 7. Note that the jump applies to the *raw* weight, before any proficiency adjustments have occurred. Note, also, that the Dutch-specific SPECLEFT is unaffected by the jump, but the General weight reflects the updated English-specific SPECLEFT value.

If we substitute these new adjusted SPECLEFT weights into Table 6, then the column for calculating SPECLEFT’s contribution to Harmony changes. For Candidate 1, which had no SPECLEFT violations, the new weights make no difference, and the contribution to harmony remains 0. For Candidate 2, there are no English violations of SPECLEFT, since the specifier is in Dutch, but the changed language-general weight still reflects the new English-specific weight. Therefore, the harmony contribution of Candidate 2’s Dutch SPECLEFT violation also changes. This new calculation is illustrated in Table 8. The values that have changed relative to the values in Table 6 have been

Input: en-sov-1	SpecLeft	HEADRIGHT	HEADLEFT	MATCHPROMPT	MAX	MATCHSWITCH	Harmony
Adjusted En ($\times .4$)	2.39	3.60	4.84	-0.80	1.68	-0.04	
Adjusted Du ($\times .6$)	1.92	3.60	4.44	3.00	2.34	0.48	
General	4.31	7.20	9.28	2.20	4.02	0.44	
1. $ S_D O_D V_D$	$0 \times \mathbf{2.39}$ $+0 \times 1.92$ $+0 \times \mathbf{4.31}$ $=0$	0×3.6 $+0 \times 3.6$ $+0 \times 7.20$ $=0$	0×4.84 $+2 \times 4.44$ $+2 \times 9.28$ $=27.44$	$0 \times -.8$ $+0 \times 3.0$ $+0 \times 2.2$ $=0$	1.35×1.68 $+0 \times 2.34$ $+1.35 \times 4.02$ $=7.70$	0×-0.04 $+0 \times 0.48$ $+0 \times 0.44$ $=0$	$-(27.44$ $+ 7.70)$ $= -35.14$
2. $V_E S_D O_E$	$0 \times \mathbf{2.39}$ $+1 \times 1.92$ $+1 \times \mathbf{4.39}$ $=\mathbf{6.23}$	2×3.6 $+0 \times 3.6$ $+2 \times 7.20$ $=21.6$	0×4.84 $+0 \times 4.44$ $+0 \times 9.28$ $=0$	$1.33 \times -.8$ $+0.67 \times 3.0$ $+2 \times 2.2$ $=5.35$	0.45×1.68 $+1.1 \times 2.34$ $+1.55 \times 4.02$ $=9.56$	0×-0.04 $+0 \times 0.48$ $+0 \times 0.44$ $=0$	$-(\mathbf{6.23}$ $+21.6$ $+5.35$ $+9.56)$ $= -$ $\mathbf{42.74}$

Table 8: The new harmony calculations after a jump has been applied to the English-specific SPECLEFT value. Changes in Harmony score calculations compared to Table 6 are in bold.

bold-faced.

Next we recalculate the predicted probability based on the new harmony score for Candidate 2. Note that since the harmony has become more negative—from -42.51 to -42.74—the new probability distribution should assign an even lower probability to Candidate 2 than before.

Recall that, under the old SPECLEFT value, the predicted probability for Candidate 1 was .9994, and for Candidate 2 .0006. We now recalculate these values, as before: After exponentiating the two harmony values, we have $e^{-35.14} = 5.4 \times 10^{-16}$ for Candidate 1, and $e^{-42.74} = 2.7 \times 10^{-19}$. Next, we add together these two exponentiated values: $5.4 \times 10^{-16} + 2.7 \times 10^{-19} = 5.5 \times 10^{-16}$. Finally, we divide by this sum the exponentiated harmony value for each candidate, to arrive at a predicted probability. For Candidate 1 the predicted probability is $\frac{5.4 \times 10^{-16}}{5.5 \times 10^{-16}} = .9995$, while for Candidate 2 this is $\frac{2.7 \times 10^{-19}}{5.48 \times 10^{-16}} = .0005$. These values are, as predicted from the changed harmony profiles, even more tilted towards Candidate 1 than before (.9995 vs. .9994), albeit only slightly.

B.2 Determining whether the fit improved

To determine whether this new weight value improves the fit of the model to the data, we must compare the match of the old probability distribution to the data with the match of the new probability distribution to the data. For the sake of argument, let’s say that we had 75 responses, and of those responses, 73 used Candidate 1, and 2 used Candidate 2. In other words, the actual usage proportions were .973 for Candidate 1, and .027 for Candidate 2.

To compare the two proposed probability distributions—the old {.9994, .0006} and the new {.9995, .0005}—to the actual proportion of responses, {.973, .027}, the algorithm calculated the Kullback-Leibler Divergences (KLD, [Kullback and Leibler, 1951](#)) for each separate prompt. KLD is an information-theoretic concept that estimates the amount of information lost when one dis-

tribution is represented in terms of another. When two distributions are exactly equivalent, the KLD is 0. When one distribution is an imperfect representation of the information contained in another, the KLD grows. Since we are interested in determining how well the model’s predicted probabilities fit the empirical data, we calculated the KLD between the two distributions, using the `KL.plugin()` command from `entropy` package (version 1.2.1, [Hausser and Strimmer, 2009](#)). We calculated the KLD using base-2 logarithms, which ensured that the information lost when representing empirical proportions as predicted probabilities was expressed in bits.

After each jump, the KLD derived from the new set of weights was compared to the KLD from the previous set of weights. We also compared the correlations between those two sets of weights. In our toy example, the KLD between the actual data of $\{.973, .027\}$ and the probabilities derived from the old weights, $\{.9994, .0006\}$, is 0.088, and the correlation is 1⁴. By contrast, the KLD between the actual data and the probabilities derived from the new weights, $\{.9995, .0005\}$, is 0.094, and the correlation is again 1. In other words, the KLD is higher under the new weights than under the old weights, indicating a poorer fit of the model predictions to the data. This poorer fit is easy to see by eye in our toy example: In both cases the predicted probability for Candidate 1 is much too high, and for Candidate 2 is much too low; but the mismatch is smaller in the case of the old weights than the new weights.

The optimization algorithm retained the new weights under two conditions: First, the new KLD had to be lower, indicating less information loss, than the KLD under the previous set of weights. Second, the correlation between the new predicted probabilities and actual proportion of uses had to be higher than the correlation under the previous set of weights. We retained the correlation as a criterion of fit because we wanted to ensure that the model was not only fitting the shape of the distribution appropriately, but was also able to match the actual proportions with their predicted probabilities.⁵ If either of these two conditions was not met, the weight values were reset to their previous values. In our toy example, the KLD under the new weights was higher, not lower, than the KLD under the old weights, so the new weights would have been discarded, and the model reset to the old weights. Then the algorithm would move on to the next constraint—here, the Dutch-specific `SPECLEFT`, and the whole process would repeat.

The algorithm was set to repeat 10,000 cycles through the 12 constraints, which resulted in 120,000 jumps.

⁴Since there are only two possible values in this toy example, the correlation could only ever be 1 or -1. In the real dataset, there were 48 possible responses being evaluated for each prompt, and so there were no such constraints on possible correlations.

⁵We also ran this algorithm using only KLD as a measure of fit, but it did not perform as well in generating probabilities that accurately matched usage proportions.

References

- Goldrick, M., Putnam, M., and Schwarz, L. (2016). Coactivation in bilingual grammars: A computational account of code mixing. *Bilingualism: Language and Cognition*, 19(5):857–876.
- Hausser, J. and Strimmer, K. (2009). Entropy inference and the James-Stein estimator, with application to nonlinear gene association networks. *Journal of Machine Learning Research*, 10:1469–1484.
- Kullback, S. and Leibler, R. A. (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.
- R Core Team (2019). R: A language and environment for statistical computing.