

UC Merced

Proceedings of the Annual Meeting of the Cognitive Science Society

Title

Constraints on Assimilation in Vowel Harmony Languages

Permalink

<https://escholarship.org/uc/item/6wz0630w>

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 12(0)

Author

Hare, Mary

Publication Date

1990

Peer reviewed

Constraints on Assimilation in Vowel Harmony Languages

Mary Hare
UC San Diego

Abstract

Over the last 10 years, the assimilation process referred to as vowel harmony has served as a test case for a number of proposals in phonological theory. Current autosegmental approaches successfully capture the intuition that vowel harmony is a dynamic process involving the interaction of a sequence of vowels; still, no theoretical analysis has offered a non-stipulative account of the inconsistent behavior of the so-called "transparent", or disharmonic, segments.

The current paper proposes a connectionist processing account of the vowel harmony phenomenon, using data from Hungarian. The strength of this account is that it demonstrates that the same general principle of assimilation which underlies the behavior of the "harmonic" forms accounts as well for the apparently exceptional "transparent" cases, without stipulation.

I. Introduction¹

The current paper proposes a connectionist processing account of certain aspects of vowel harmony in Hungarian. The paper has two interrelated goals. First, it offers an explanatory account of the behavior of the so-called *transparent* vowels in that language. Second, this account relies crucially on a connectionist theory of sequential processes: thus to the extent it succeeds it demonstrates the utility of connectionist models as an explanatory tool in the study of linguistic phenomena.

The paper is organized in the following manner: I first review some facts about the vowel harmony process in Hungarian which present difficulties to analysis. Second, I introduce the model of sequential processes developed by Jordan (1986). The core of the paper then involves a series of parametric studies, whose aim is to determine the conditions on assimilation in a network of this type. Having established what factors constrain assimilation in the sequential network, I return to the Hungarian data, and show that the interaction of these same factors predicts the correct pattern of behavior for both harmonic and transparent vowels in that language.

II. The data

The Hungarian vowel system is as shown below. This is a seven vowel system, and each vowel has a long counterpart which is phonemic. Notice that there is a round - nonround distinction among the non-low vowels, and that while /e:/ is a mid vowel, /e/ is analyzed as being low.

front:		back:
unrounded:	rounded:	
i i:	ü ü:	u u:
e:	ö ö:	o o:
e		a

Hungarian exhibits front-back harmony: in general roots contain only front or only back vowels, and suffix vowels alternate to agree in backness with those of the root. The following data exemplify the harmony phenomenon. These are examples of consistent front- and back-vowel roots, followed by the dative suffix. Note that after a front root, the suffix takes the form *nek* while after a back root the same suffix is realized as *nak* (data from Vago 1979).

(1) Front roots:

iker 'twin'	iker-nek	'twin-DAT'
tükör 'mirror'	tükör-nek	'mirror-DAT'

(2) Back roots:

varos 'city'	varos-nak 'city-DAT'
kapu 'gate'	kapu-nak 'gate-DAT'

There are a number of exceptions to the pattern of consistent harmony within roots. The most important exception involves the class of non-low front unrounded vowels (/i/, /i:/, /e:/, and on some accounts /e/). As the following examples demonstrate, these can also appear in the same root as a back vowel. In these cases the back vowel of the root, regardless of its position, determines the backness quality of the suffix vowel.

(3)

bika 'bull'	bika-nak 'bull-DAT'
izom 'tendon'	izom-nak 'tendon-DAT'

(4)

kosci 'carriage'	kosci-nak 'carriage-DAT'
taxi	taxi-nak 'taxi-DAT'

Note, however, that in certain environments the front vowels *do* determine the backness value of the suffix. This is the case if the root contains only front vowels, as in (1), or if the root ends in a sequence of such vowels, as in the (borrowed) forms shown below (data from Kontra and Ringen 1987).

(5)

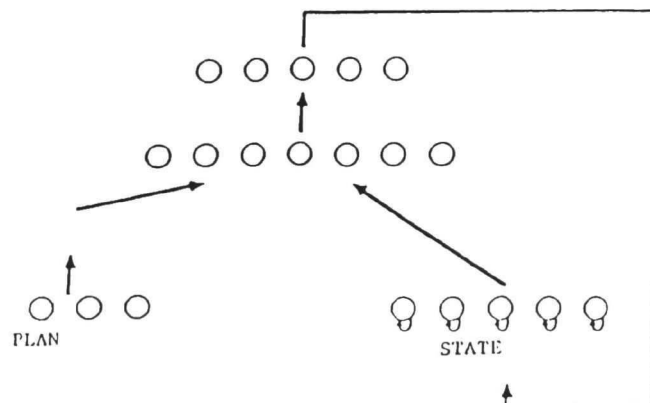
aspirin	aspirin-nek
	* aspirin-nak
bronkitis	
	* bronkitis-nak

The problem, then, is that an identical vowel may behave harmonically in one environment, while violating harmony in another. This complication has often been dealt with in the literature by positing a number of different sources for the segment in question, and allowing the harmonic-nonharmonic distinction to follow from this (Clements 1986, van der Hulst 1985, Ringen 1989, among others). One drawback to such an approach is that there is no reason to establish these differences in derivational source except to distinguish between harmonic and non-harmonic behavior: there is no other behavior in the phonology of the language that motivates it. A preferable account would be one in which these differences in behavior follow from general conditions on the model. In what follows I will use a connectionist model of assimilation to suggest one such account.

III. Sequential processing in the connectionist framework

The account that is being developed here relies on the theory of sequential processes developed by Michael Jordan. Jordan (1986) describes an interesting series of models of coarticulation effects, using a recurrent connectionist network which learns to produce an ordered sequence of output patterns in response to a given input. The network is illustrated in Diagram I.

(6) Diagram I



(Jordan 1986)

These models involve at least three layers of processing units: input, internal (or *hidden*), and output. Activation passes from the input to the output along weighted connections. Input to the model consists of two parts. The first, labeled the *plan*, is an arbitrary vector that triggers the production of a given sequence. In addition, the *state* of the system (that is, the current output) is fed back over fixed connections and constitutes part of the input at the next cycle. This serves as a temporal context and aids the system in learning what part of the sequence is the next to be produced. Learning is accomplished through the back propagation of error algorithm. After each input is presented, the output that results is compared to the desired output, and the discrepancy between the two is computed. This discrepancy is the *error* on that pattern. The weights on the connections are modified slightly to minimize this error. This process is repeated until some criterion of acceptability is reached.

In the simulations to be discussed here, as in Jordan's coarticulation model, output at any given time consists of a single phoneme represented by a vector corresponding to a distinctive feature description. A word or other longer sequence is represented over time as a string of phonemes on successive output cycles. One interesting property of this network is that particular features of a phoneme can be left unspecified for any value. This is accomplished by having no error signal propagated back from that unit. Instead of learning to match a particular teacher, the unit picks up its specification from some other pattern in the sequence.

In what follows I will use the term *assimilation* to refer to the tendency of an unspecified output unit (hereafter a *don't care* unit) to take on a value influenced by one of its neighbors in time. Jordan shows that outputs tend to follow as smooth a trajectory as possible: thus a don't care unit might be expected to assimilate most strongly to its immediate temporal predecessor. In certain cases, however, the don't care unit ignores its immediate predecessor, and takes on a value close to that of an earlier pattern in the sequence. The question, then, is to determine what factors influence the choice of assimilatory trigger. Note that this exactly the problem in the Hungarian data, as well.

IV. Conditions on assimilation in the sequential network²

The following set of simulations were designed to test the hypothesis that the similarity between vowels (in a sense which will be made precise below) is a crucial factor in determining the choice of assimilatory trigger. Stimuli were output sequences as in the example below.

(7)

1	0 1 0 0 0 1 0
2	1 0 1 1 1 0 1
3	* 1 0 0 0 1 0

Each output was a seven-bit distributed pattern, and for each plan the network learned to produce a three-pattern sequence whose members will be referred to as 1, 2, and 3. In each sequence the first two patterns (1 and 2) are specified for all seven units, while the third (3) has one don't care unit, in initial position in the string. (The don't care unit is indicated by the asterisk.) 1 and 2 have opposing values on this first unit.

Sets of patterns were devised in which the final two lines (2 & 3) were held constant with certain number of units in common. This measure of 'units in common' is referred to as the 'hamming distance' between 2 and 3, and is a measure of vector similarity. The first line in the sequence (line 1) was varied in similarity to the other two by manipulating the hamming distance between them. This was done in the following way. The pattern given in (7) was the first 3-line sequence in one such set. Here 2 & 3 have opposing values on the last six bits, while 1 and 3 are identical. In the second sequence of this set (example 8) 2 and 3 remain unchanged while 1 is varied to differ from 3 on one unit. In the third sequence, (given in 9) 1 and 3 differ on two units.

(8)

1	0 0 0 0 0 1 0
2	1 0 1 1 1 0 1
3	* 1 0 0 0 1 0

(9)

1	0010010
2	1011101
3	*100010

This process was repeated, steadily decreasing the similarity between 1 and 3 until the set consisted of seven 3-line sequences. Note that by *similarity* I am speaking of hamming distance, a measure of overall vector similarity, and not simply the presence of similar values on any single unit.

a. Training

These sequences served as teaching output to the network described above. This network was trained on each sequence for 2000 iterations, where an iteration is one presentation of one pattern. In learning to produce the sequences, the network also assigns a value to the don't care unit. This unit is expected to simply maintain the value of the previous pattern; the goal of these simulations is to determine under what conditions the *don't care* unit reverts to the value of the first pattern instead. After 2000 iterations, the training was stopped and the actual output was examined to determine the value taht the don't care unit had taken on.

b. Results

Results from the first set of simulations show that the don't care unit in 3 consistently assimilates to the corresponding unit in 1 when these two patterns are most similar. Thus although the default case in the Jordan network is for a don't care unit to maintain the value of the immediately previous output, this unit does not exclusively influence the result. If the pattern two time steps back is strongly similar to the target, the don't care unit will take a value nearer the corresponding unit in that pattern instead.

These results are given in Graph I, which should be read as follows. Distance along the x-axis measure the similarity between patterns 1 and 3 - that is, between the first and third line of each sequence. The y-axis gives the activation level taken on by the don't care unit after 2000 iterations of learning. The boldface bar gives the average activation over a number of trials, while the line through each bar marks the limits of the variability. As the graph shows, as patterns 1 and 3 become more alike, there is a corresponding increase in the influence of 1 as assimilatory trigger.

These results are typical of a pattern which emerged over a number of pattern sets. The second graph gives the results from a second set of sequences, in which the second pattern was held constant at a hamming distance of 5 units from the target, while the first pattern was varied as before. The only difference between the simulations reported above and this set is that 2, the second pattern in the sequence, is here slightly more similar to 3, the target. In this set, the same pattern of results emerges.

Notice also that the second graph begins with an additional column, where all values are clustered near 0, indicating low assimilation to 1 and high assimilation to 2. This is the output from what will be referred to at the identity condition, where 1 and 2 are not only equal in hamming distance from 3, but are identical. This result shows that when the two potential trigger patterns are identical or nearly so, the target pattern assimilates to the second of the two in all cases. This in not surprising, given the model. A basic property of these networks is that, similar inputs produce similar outputs. Since here temporal context is treated as part of the input, patterns learned in very similar temporal contexts are expected to exhibit very similar behavior.

These simulations were repeated under a number of conditions, with the hamming distance between 2 and 3 progressively decreased. The same pattern of results continued to appear, although in an increasingly attenuated form. Consistently, the influence of the first pattern of the sequence is strongest when it is most similar to the target.

V. Analysis of Hungarian vowel harmony

This pattern of results shows that in a processor of this sort the similarity structure of output strings across time influences assimilatory behavior. Returning to the Hungarian data, let us consider how the facts of the transparent vowels of that language agree with the behavior of the sequential network.

Here I modeled the behavior of a series of Hungarian words in the same assimilation task. In this case the output sequences were not arbitrary bit strings chosen only for their similarity structure, but vectors corresponding to distinctive feature representations of phonemes. The features used to represent the vowels were *back*, *high*, *low*, and *round*.

(10) Vowel Code:

i	0 1 0 0	ö	0 0 0 1
e:	0 0 0 0	u	1 1 0 1
e	0 0 1 0	o	1 0 0 1
ü	0 1 0 1	a	1 0 1 0

Words being modeled were represented only by their vowels. Each sequence consisted of vectors representing two or more root vowels specified for all features, and a third which represented the vowel of the dative suffix. This was given as a low unrounded vowel unspecified for [back]. As before, lack of specification equated to a don't care condition on the relevant unit.

(11) iker - nek

i	0 1 0 0
e	0 0 1 0
e	* 0 1 0

Each pattern was learned separately, as before, for 2000 iterations. At this point the underspecified vowel had taken on a value for [back] influenced either by its immediate predecessor, or by an earlier member of the sequence.

To summarize the results of the earlier simulations, a don't care unit will generally maintain the value on the corresponding unit in the previous output. However, if the immediate predecessor is very dissimilar to the target, it is less likely to trigger assimilation. If the antepenultimate member of the sequence shows a strong similarity to the target it instead will be chosen as the trigger, and the penultimate member will be ignored. Furthermore, the similarity between the two potential triggers plays a role. If these two are identical, or markedly similar, the target will assimilate to the second of the two in all cases. The current simulations look at how these results explain the real language data.

In the data presented below, the expected (or *teacher*) output is given first, followed by the actual output of the network after 2000 iterations of training. The unspecified unit in the teacher is represented with an asterisk (*), and the corresponding output is given in boldface.

For the harmonic roots, both potential triggers have the same value for [back]. This value is straightforwardly maintained onto the don't care unit as a result of the smoothness constraint.

(12) iker - nek

i	0 1 0 0	0.053	0.824	0.173	0.059
e	0 0 1 0	0.036	0.149	0.853	0.032
e	* 0 1 0	0.030	0.093	0.910	0.023

(13) kapu - nak

a	1 0 1 0	0.951	0.099	0.911	0.086
u	1 1 0 1	0.962	0.817	0.171	0.833
a	* 0 1 0	0.947	0.120	0.886	0.111

In the mixed roots, the examples all contain a high front vowel in one syllable and a back vowel elsewhere. In these examples, the vectors representing both the suffix vowel and the back root vowel differ significantly from the root vowel *i*. Thus it is expected that even when *i* immediately precedes the underspecified vowel, it will exert little assimilatory influence. In addition, the first pattern in the string is strong similar to the target, and so is expected to have a strong influence. This is in fact the case, both in the simulation and in the real language data.

(14) taxi - nak

a	1 0 1 0	0.811	0.106	0.878	0.032
i	0 1 0 0	0.231	0.781	0.240	0.035
a	* 0 1 0	0.700	0.200	0.785	0.029

(15) bika - nak

i	0 1 0 0	0.134	0.877	0.130	0.050
a	1 0 1 0	0.845	0.110	0.883	0.025
a	* 0 1 0	0.870	0.071	0.930	0.029

However, the situation changes when the root contains a sequence of non-low front vowels. In these examples the target is preceded by a sequence of identical vectors. Here the identity of temporal context is the strongest factor, and the don't care unit is expected to assume the value of its immediate predecessor. Again, this behavior parallels the Hungarian facts.

(16) aspirin - nek

a	1 0 1 0	0.758	0.252	0.750	0.060
i	0 1 0 0	0.138	0.794	0.198	0.014
i	0 1 0 0	0.175	0.671	0.330	0.026
e	* 0 1 0	0.174	0.633	0.372	0.032

VI. Conclusion

To summarize, although the expected pattern in Hungarian is that all vowels of a word will agree in backness, certain front vowels in some environments respect this pattern, and in other environments do not. This more complex behavior is a function of both segmental identity and temporal context. Here I have suggested a processing treatment of the Hungarian facts which predicts harmonic behavior for these vowels on the basis of the overall similarity relationships among the vowels of the word.

A number of factors argue in favor of such an analysis. First, it offers a simpler and more explanatory account of Hungarian data. The vowels which exhibit transparent behavior, and the environments in which this behavior will change, must be stipulated arbitrarily under traditional accounts, while both follow automatically from the account proposed here. Second, this account makes strong claims about the existence of possible harmony patterns. As was demonstrated above, it is a general property of the sequential network that the assimilation process is sensitive to similarity in the temporal context. This predicts substantive constraints on what patterns of harmony may or may not exist. Whether these predictions can be maintained as a general principle requires further research, but available data suggest them to be correct.

Finally, although this account suggests that modifications of the autosegmental treatment of harmony are necessary, it is heavily influenced by the autosegmental notion of assimilation as the spread of a value for a particular phonetic feature. As the simulations demonstrate, a properly constrained treatment of temporal spread correctly predicts those harmonic patterns which exist in Hungarian, while failing to produce non-attested patterns. This result argues strongly in favor of the use of processing models as sources of constraint and explanation which can potentially enrich linguistic theory.

Notes

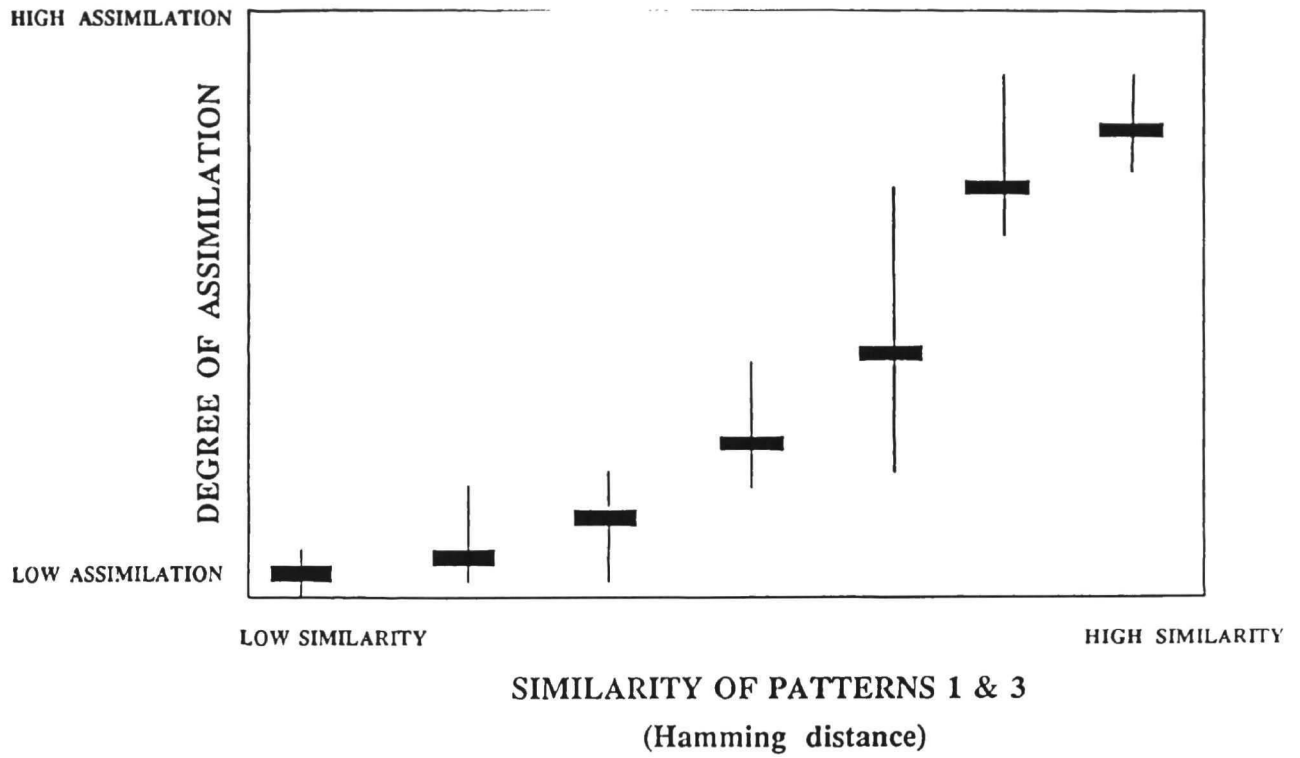
¹ I am grateful to Jeff Elman, Rob Kluender, Steve Poteet, Robert Port, George Lakoff, Sanford Schane, Gary Cottrell, Ann Thyme, Errapel Mejias-Bikandi and Kathleen Carey for useful comments and discussion.

² Results reported here are from a recurrent network with two input (plan) units, seven output and consequently seven state units, and six hidden units. The learning rate in simulations was 0.1; *mu* (the multiplier on the recurrent connections from each state unit to itself) was 0.6, and momentum was set to 0.

Bibliography

- Archangeli, Diane. 1984. *Underspecification in Yawelmani Phonology and Morphology*. MIT PhD dissertation.
- Archangeli, Diane and Douglas Pullyblank. 1986. *The Content and Structure of Phonological Representations*. ms.
- Clements, George N. 1976. *Vowel harmony in non-linear generative phonology*. IULC.
- Goldsmith, John. 1976. *Autosegmental Phonology*. PhD dissertation, MIT. Published 1979, New York: Garland Press.
- Goldsmith, John. 1985. *Vowel harmony in Khalkha Mongolian, Yaka, Finnish, and Hungarian*. *Phonology Yearbook* 2. 251-275.
- Goldsmith, John. 1989. *Autosegmental and Metrical Phonology*. Basil Blackwell.
- Hulst, H. van der. 1985. *Vowel harmony in Hungarian: A comparison of segmental and autosegmental analyses*. In van der Hulst and Smith 1985.
- Hulst, H. van der and Norval Smith (eds) 1985. *Advances in nonlinear phonology*. Dordrecht: Foris.
- Jensen, J.T. 1978. *A reply to "Theoretical implications of Hungarian vowel harmony"*. *Linguistic Inquiry* 9, 89-97.
- Jordan, Michael. 1986. *Serial Order: a Parallel Distributed Processing Approach*. Institute for Cognitive Science Report #8604. UC San Diego.
- Kontra, M. and Catherine Ringen. 1987. *Stress and harmony in Hungarian loanwords*. in Redei 1987.
- McClelland, James L., David E. Rumelhart, and the PDP Research Group. 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 2: Psychological and Biological Models*. Cambridge, Mass.: MIT Press/Bradford Books.
- Pullyblank, Douglas. 1988. *Underspecification, the feature hierarchy, and Tiv vowels*. *Phonology* 5.2.
- Redei, K. (ed) 1987. *Studien sur Phonologie und Morphonologie der uralischen Sprachen*. *Studia Uralica*, Wein: Verbund der wissenschaftlichen Gesellschaften Osterreichs. 81-96.
- Ringen, Catherine. 1988. *Transparency in Hungarian Vowel Harmony*. *Phonology Yearbook* 5.2.
- Rumelhart, David E., James L. McClelland, and the PDP Research Group. 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*. Cambridge, Mass.: MIT Press/Bradford Books.
- Vago, Robert M. 1976. *Theoretical implications of Hungarian vowel harmony*. *Linguistic Inquiry* 7, 243-263.

Graph I



Graph II

