Winning isn’t everything. It’s the only thing.

Vince Lombardi

In determining the correct – optimal – parse of an input, as the constraint hierarchy is descended, each constraint acts to disqualify competitors with absolute independence from all other constraints. A parse found wanting on one constraint has absolutely no hope of redeeming itself by faring well on any or all lower-ranking constraints.

Prince and Smolensky (1993:78)

CHAPTER 1

INTRODUCTION

In this dissertation I propose a new way to think about how EVAL, the evaluative component of an Optimality Theoretic grammar, works. In particular, I propose two changes in the way that EVAL is usually viewed in classic OT (Prince and Smolensky, 1993): (i) First, I propose a change in the information structure that EVAL imposes on the candidate set. In classic OT, EVAL is seen as defining only a two-level distinction in the candidate set – i.e. EVAL distinguishes between the best candidate and the set of losers, but does not impose any structure on the set of losers. I argue that EVAL imposes a harmonic rank-ordering on the complete candidate set – i.e. also the losers are ordered from best to worst. Rather than just a two-level ordering, EVAL imposes a multi-level harmonic ordering on the candidate set. I therefore refer to the model of EVAL that I propose as the "rank-ordering model of EVAL". (ii) Secondly, I argue that we should expand the comparative powers of EVAL. In classic OT EVAL is assumed to compare candidate sets of only one kind, namely sets generated by GEN as candidate outputs for a specific input. (I will refer to these sets as "generated comparison sets".) Candidates in a generated comparison set are all morphologically related to each other via a shared input. However, I will show that EVAL can compare any set of candidates, even candidates that are not related to each other via a shared input (i.e. EVAL can also compare with each other mappings such as the following: /in_1/ → [$cand_1$] and /in_2/ → [$cand_2$], where /in_1/ and /in_2/ are morphologically unrelated).

The motivation for this alternative view of EVAL is two-fold. First, I will show that this alternative view of EVAL does not require any changes to the architecture of EVAL in a classic OT grammar. Even in a classic OT grammar EVAL generates the information necessary to impose a harmonic rank-ordering on the complete candidate set, and even in classic OT EVAL has the ability to evaluate non-generated comparison sets. I am not really proposing a change to the architecture of a classic OT grammar. I am rather pointing out two previously unnoticed and/or unappreciated features of an OT grammar. Secondly, I will show that the different conceptualization of EVAL extends the empirical coverage of an OT grammar. Specifically, it enables us to account for non-categorical phenomena. I will discuss two kinds of non-categorical phenomena in detail in this dissertation, namely variation in production (more than one grammatical pronunciation for the same word) and the processing of non-words (well-formedness judgments and lexical decision).

The rest of this introductory chapter is structured as follows. In §1 I will discuss the alternative view of EVAL that I am proposing in more detail. In §2 I will then present a brief illustration of how this alternative view of EVAL can be used to account for variation in production, and in §3 I will show how it can be used to account for the processing of non-words. Finally, in §4 I will explain the structure of the rest of the dissertation.

1. Theoretical preliminaries

I am proposing two innovations to the way that we standardly think about an OT grammar. Neither of the innovations requires a change to the architecture of the grammar.

One of the innovations makes use of information that a classic OT grammar generates but that is usually considered irrelevant in OT literature, namely about the relationships between the non-optimal candidates in the candidate set. The other entails an extension of the comparative powers of EVAL so that it can compare candidates from different, morphologically unrelated inputs. Comparing such morphologically unrelated forms is something that even the EVAL of classic OT could do. However, this ability of EVAL was never appreciated in classic OT. Each of these two extensions is discussed in more detail below.

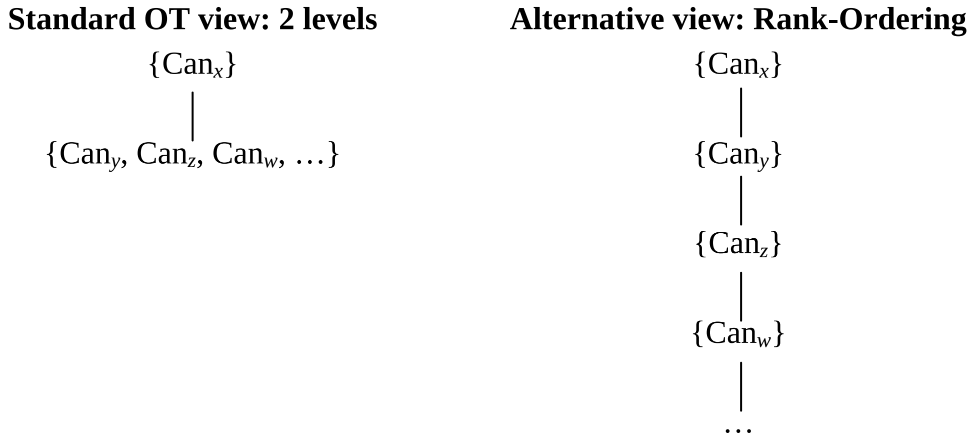
1.1 On being a better (or a worse) loser: a rank-ordering model of EVAL

OT is a theory of winners. EVAL makes only one distinction in the candidate set, namely between the winning candidate and the mass of losers. There is no claim made about the relationships between the losers. There is no concept such as being a better or a worse loser.

This standard view of OT is held in spite of the fact that EVAL has the power to make finer grained distinctions in the candidate set. If we were to remove the optimal candidate from a candidate set, and submit only the set of losers to EVAL, then EVAL will again identify the one loser that is better than all the rest. This best loser can then be removed, and we can allow EVAL to compare only the smaller set of remaining losers, again identifying the best one from this smaller set. In fact, this process can be repeated for as long as there are still candidates left, and we can therefore rank-order the full candidate set in this way.

In classic OT the assumption is that EVAL imposes only a two-level harmonic ordering on the candidate set (the optimal candidate against the rest). Alternatively, we can also entertain the possibility that EVAL imposes a harmonic rank-ordering on the full candidate set. These two views about the information structure that EVAL imposes on the candidate set can be represented graphically as in (1). Candidates appearing higher are more harmonic relative to the constraint ranking.

(1)



Standard OT view: 2 levels

Alternative view: Rank-Ordering

In classic OT the losers are lumped together in an amorphous set. It does not matter whether you are the second best candidate or the worst candidate – all that counts is that you are not the winner. In the rank-ordering model of EVAL this is not the case. Even the losers are ordered with respect to each other. It is now possible to be a better or a worse loser – i.e. being the second best candidate is now something that has meaning.

Under the standard assumption that every constraint evaluates every candidate, the information necessary to rank-order the full candidate set is automatically generated by an OT grammar. It is not necessary to make any additions to the way that an OT grammar works in order to get this information. The assumption in classic OT that EVAL imposes only a two-level ordering on the candidate set is therefore not a theoretical necessity. It is rather the case that this information is available, but that classic OT assumes it to be irrelevant. If this information were really irrelevant, then an OT grammar would have been much too powerful, generating a mass of irrelevant information. I will therefore argue for the opposite, namely that this information is relevant and that it available to and accessed by language users in non-categorical phenomena. This lends strong support to the general architecture of an OT grammar.

Since the alternative view about the output of the grammar is that EVAL imposes a rank-ordering on the full candidate set, I will refer to this model as the “rank-ordering model of EVAL”.

1.2 Extending the comparative powers of EVAL

The basic function of EVAL is to compare a set of candidates in terms of their relative harmony. In classic OT it is assumed that the set that EVAL compares is the set of candidate output forms for a single input – i.e. the candidates generated by GEN for some input. However, there is nothing in the way EVAL functions that requires the comparison set to be of this kind. EVAL is blind to the origin of the candidate set that it compares, and in principle EVAL can compare any set of candidates, whether they all come from the same input or each from a different input.

To make this discussion more concrete, consider a language *L* that tolerates codas, i.e. with the ranking ||{MAX, DEP} o NOCODA||. Suppose that *L* had only two lexical entries, namely /pata/ and /kispa/. Since *L* tolerates codas, both of these underlying representations should be mapped faithfully onto the surface. This is easily confirmed by allowing EVAL to evaluate the candidate sets generated by GEN for each of /pata/ and /kispa/. This is the kind of comparison that is traditionally done in OT. In (2) I show the tableaux for these comparisons.

(2) **EVAL evaluates generated comparison sets**

a. **/pata/ → [pa.ta]**

/pata/	MAX	DEP	NOCODA
L pa.ta			
pa.tak		*!	*
pat	*!		*

b. **/kispa/ → [kis.pa]**

/kispa/	MAX	DEP	NOCODA
L kis.pa			*
ki.si.pa		*!	
ki.pa	*!		

In (2a) the faithful candidate does not violate any constraint, and is therefore necessarily selected as output. The tableau in (2b) confirms that *L* will not avoid codas by either epenthesis or deletion.

But this is not the only informative comparison that we can make. Even though *L* tolerates codas (as shown in (2b)), it is still the case that a form with a coda is more marked than a form without a coda – i.e. although both [pa.ta] and [kis.pa] are possible words of *L*, [kis.pa] is more marked because it earns a violation of NOCODA while [pa.ta] does not. This intuition can be expressed formally by allowing EVAL to compare these two candidates. This is shown in the tableau in (3).

(3) **Comparison between the faithful candidates**

		MAX	DEP	NOCODA
2	/kispa/ → [kis.pa]			*
1	/pata/ → [pa.ta]			

Two things can be remarked about the tableau in (3). First, although these two candidates are not morphologically related to each other via a shared input, they can be compared in a straightforward manner. No additions have to be made to the way in which EVAL works in order to compare two such morphologically unrelated candidates. Secondly, this tableau supplies us with information that we would not have access to if we considered only the generated comparison sets of classic OT. That a possible word with a coda is somehow less well-formed than a possible word without a coda is not expressed in the tableaux in (2). These tableaux show only that words with and without codas are allowed in *L*. The difference between words with and without codas can only be expressed in comparisons such as that done in tableau (3). Of the two forms compared, the form without the coda is rated better by EVAL. (In this tableau I indicate this with Arabic numerals – the 1 next to the second candidate indicates that it is rated best, the 2 next to the first candidate that it is rated second best. More on these conventions follows in §2.)

EVAL therefore has the power to compare candidates that are not related to each other via a shared input, and comparisons like these can provide us with information that is not otherwise available. I will show in the rest of this dissertation how information from such comparisons is used in accounting for non-categorical phenomena.

What are all the comparison sets that EVAL can evaluate? In principle there is no limit on the sets of candidates that EVAL can compare. If we collect together in one set the candidates that GEN generates for all inputs, EVAL can compare any subset of this large set. In (4) I formalize what the sets are that EVAL can evaluate.

(4) **Comparison sets**

Let *Input* = {in_1, in_2, … in_n} be the set of all possible inputs, and GEN(in_i) the set of candidates generated by GEN for input in_i.

Then the set of all possible candidates is the arbitrary union of the sets generated by GEN for all possible inputs, i.e. $\cup_i$ GEN(in_i) for all in_i in *Input.*

EVAL can compare any subset of candidates from $\cup_i$ GEN(in_i). The set of possible comparison sets is therefore the powerset of this set, i.e. $\wp$($\cup_i$ GEN(in_i)).

The set generated by GEN for some single input in_j is a special kind of comparison set. It is namely that member of $\wp$($\cup_i$ GEN(in_i)) that includes all and only the members of GEN(in_j). In order to distinguish this sub-class of possible comparison sets from the rest of the possible comparison sets, I will refer to these special comparison sets as *generated comparison sets* – to emphasize that these are the sets generated by GEN for specific inputs. Comparison sets that do not qualify as generated comparison sets will simply be called *non-generated comparison sets*.

Consider again language *L* from above. Recall that we assumed that *L* has only two lexical forms, /pata/ and /kispa/. For *L* the set *Input* is then {/pata/, /kispa/}. The set of all possible candidates in *L* is the set (GEN(/pata/) ∪ GEN(/kispa/)), and EVAL can compare any subset of (GEN(/pata/) ∪ GEN(/kispa/)), i.e. any member of the powerset $\wp$(GEN(/pata/) ∪ GEN(/kispa/)). In (5) I list a few of comparison sets for *L* that EVAL can evaluate.

(5) **Comparison sets in *L***

a. **Generated comparison sets**

GEN(/pata/) = {/pata/ → [pa.ta],
/pata/ → [pa.ta.ka],
/pata/ → [pat], …}

GEN(/kispa/) = {/kispa/ → [kis.pa],
/kispa/ → [ki.si.pa,
/kispa/ → [ki.pa], …}

b. **All possible candidates**

GEN(/pata/)
∪ GEN(/kispa/) = {/pata/ → [pa.ta],
/kispa/ → [kis.pa],
/kispa/ → [ki.si.pa],
/pata/ → [pat], …}

c. **Non-generated comparison sets**

Set$_1$ = {/pata/ → [pa.ta], /kispa/ → [kis.pa]}
Set$_2$ = {/pata/ → [pat], /kispa/ → [ki.pa]}
Set$_3$ = {/pata/ → [pa.tak], /kispa/ → [ki.si.pa]}
Etc.

EVAL can compare any of the sets from (5a) or (5c). The sets in (5a) are the generated comparison sets of classic OT. In (2) above I showed how EVAL would compare the candidates from these two sets. The first of the non-generated comparison sets in (5c) consists of the two faithful candidates from GEN(/pata/) and GEN(/kispa/). This set was evaluated by EVAL in (3) above.

2. Applying the rank-ordering model of EVAL to phonological variation

In this dissertation I will argue that EVAL imposes a harmonic rank-ordering on the full candidate set – as shown in (1) above. The output of the grammar therefore contains more information than just what the best candidate is. It also contains a wealth of information about how the non-optimal candidates (the losers of classic OT) are related to each other. I will argue that this enriched information is available to language users, and that they access it *inter alia* in non-categorical phenomena. In this section I will provide a brief discussion of how this enriched information can be used to explain variation in production.

There is a large body of literature on phonological variation, both within sociolinguistics and within formal linguistic theory (e.g. Anttila, 1997, 2002, Anttila and Cho, 1998, Anttila and Revithiadou, 2000, Anttila and Fong, 2002, Guy, 1980, 1981, 1991a, 1991b, 1994, 1997, Guy and Boberg, 1997, Guy and Boyd, 1990, Hinskens, 1996, Hinskens *et al.*, 1997, Kiparsky, 1993, Nagy and Reynolds, 1997, Patrick, 1991, Reynolds, 1994, Reynolds and Sheffer, 1994, Silva, 1988, 1991, 1993, 1997, 1998). One thing that stands out from all of this literature is an acknowledgement that phonological variation is at least partially grammatically determined. If there is more than one pronunciation possible for a specific word, then grammar plays a role in determining how likely each variant is to be observed. In order to account for this aspect of phonological variation, it is therefore necessary to rely on grammar.

I will argue that the rank-ordering model of EVAL can be used to account for the role that grammar plays in phonological variation. In this section I will give a brief illustration of how this can be done. In this discussion I will use variable [t, d]-deletion in

Jamaican English as an example. The purpose of this discussion is to explain the theoretical assumptions that I make, and not to provide a full account of [t, d]-deletion in Jamaican English. For a detailed discussion of this phenomenon, see Chapter 5 below. The rest of this section is structured as follows: In §2.1, I give a small sample of the data on [t, d]-deletion in Jamaican English. In §2.2, I then show how the rank-ordering model of EVAL can be used to account for these data.

2.1 [t, d]-deletion in Jamaican English

In English, a [t, d] that appears as last member of a word-final consonant cluster is subject to variable deletion – i.e. a word such as *west* can be pronounced as either [wɛst] or [wɛs]. This phenomenon has been studied extensively over the past four decades and I discuss it in detail in Chapter 5. Here I will discuss only one aspect of this process in Jamaican English as reported by Patrick (1991).

Patrick reports that the likelihood of [t, d]-deletion depends *inter alia* on what follows on the [t, d]. [t, d] followed by a consonant is more likely to delete than [t, d] followed by a vowel. The table in (6) contains the relevant data (Patrick, 1991:181).

(6) **[t, d]-deletion in Jamaican English**

	Pre-C	Pre-V
n	1,252	793
% deleted	85	63

When a word like *west* is followed by a consonant, as in *west bank*, there is an 85% percent probability that it will be pronounced without the final [t] as [wɛs bæŋk], and a 15% probability that it will be pronounced with the final [t] as [wɛst bæŋk]. However, when it is followed by a vowel, as in *west end*, there is a 63% that the final [t] will not be pronounced as in [wɛs ɛnd], and a 37% chance that the final [t] will be pronounced as in [wɛst ɛnd].

In this phenomenon we are dealing with two variants, the retention candidate and the deletion candidate. There are two aspects of this variation pattern that need to be accounted for. (i) *Intra-contextual variation*. For any given input, we have to account for the fact that one of the two variants occur more frequently than the other. In both pre-vocalic and pre-consonantal context, the deletion variant occurs more frequently in Jamaican English. (ii) *Inter-contextual variation*. Although deletion is preferred over retention both pre-vocalically and pre-consonantally, it is still the case that pre-consonantal context shows higher deletion rates than pre-vocalic context. In §2.2 I explain how each of these two aspects of the variation phenomenon is accounted for in the rank-ordering model of EVAL.

2.2 Accounting for the Jamaican pattern in a rank-ordering model of EVAL

2.2.1 Intra-contextual variation

For any given input, EVAL evaluates the generated comparison set and imposes a harmonic rank-ordering on this set. Language users can potentially access all the candidates in this rank-ordering. However, the accessibility of a candidate depends directly on how high it occurs in the rank-ordering. The candidate that occupies the highest slot in the rank-ordering is most accessible and is most likely to be selected as output. This will therefore be the most frequently observed variant. The candidate that occupies the second slot in the rank-ordering, is second most accessible, and will the second most frequent variant, etc.

In both pre-consonantal and pre-vocalic context the deletion candidate is the most frequently observed candidate. (I will use ∅ to stand for the deletion candidate in the discussion below.) EVAL therefore has to rate the deletion candidate better than the retention candidate in both of these contexts, i.e. EVAL needs to impose the following rank-ordering on these two candidates |∅ ™ t/d|. This can be achieved only if the highest ranked constraint that distinguishes between the deletion and the retention candidate is a constraint that favors deletion over retention.[1] The deletion candidate obviously violates the anti-deletion constraint MAX. I will assume the existence of the markedness constraints in (7a) that militate against the retention of [t, d] in pre-vocalic and pre-consonantal position. I also assume the fixed ranking in (7b) between these two constraints. For motivation of these constraints and this ranking, see Chapter 5 §1.2.1.

(7) a. **Markedness constraints**

***Ct#C**: A word-final [t, d] is not allowed if it is both preceded and followed by a consonant.

***Ct#V**: A word-final [t, d] is not allowed if it is preceded by another consonant and followed by a vowel.

b. **Ranking**

||*Ct#C o *Ct#V||

[1] The idea of constraints favoring or disfavoring a candidate comes from Samek-Lodovici and Prince (1999). Let $C(x)$ represent the number of violations constraint C assigns to candidate x, and K the set of all candidates under consideration. For some candidate *Can* to be favored by constraint C, the following statement must then be true: $\neg\exists\ k \in K\ (C(k) < C(Can))$. Conversely, for some candidate *Can* to be disfavored by constraint C, the following statement must be true: $\exists\ k \in K\ (C(k) < C(Can))$.

Since we need the deletion candidate to be preferred over the retention candidate in both contexts, we need the markedness constraints to outrank MAX, i.e. ||*Ct#C o *Ct#V o MAX||. The tableaux in (8) show how EVAL will evaluate the generated comparison sets for a pre-vocalic and pre-consonantal input with this ranking.

(8) **Generated comparison sets evaluated**

a. **Pre-consonantal context**

/…Ct # C/	*Ct#C	*Ct#V	MAX
1 ∅			*
2 t	*		

Output of EVAL

L ∅ MAX
|
L t *Ct#C

b. **Pre-vocalic context**

/…Ct # V/	*Ct#C	*Ct#V	MAX
1 ∅			*
2 t		*	

Output of EVAL

L ∅ MAX
|
L t *Ct#V

In (8a) we see that the deletion candidate violates only MAX while the retention candidate violates only *Ct#C. Because MAX is the lower ranked of these two constraints, the deletion candidate is rated better by EVAL. This is indicated in two ways. First, by the Arabic numerals next to the candidates. The numeral 1 next to the deletion candidate indicates that this is the candidate rated best by EVAL – i.e. this is the candidate that occupies the highest slot in the rank-ordering that EVAL imposes on the candidate set, and also the optimal candidate of classic OT. The numeral 2 next to the retention candidate indicates that it is rated second best by EVAL. Below the tableau I also give a graphic representation of the rank-ordering that EVAL imposes on the candidate set. Candidates that appear higher on this ordering are rated better by EVAL. Every candidate is also indexed with the highest ranked constraint that disfavors the candidate. I also indicate each of the candidates that are observed as outputs by the familiar pointing hand **L** in this graphic representation.

The claim that I make in this dissertation is that language users have access to the full candidate set via the rank-ordering that EVAL imposes on the candidate set. Unlike in classic OT, the language user can therefore also access candidates other than the best or optimal candidate. This explains why both the best candidate (deletion) and the second best candidate (retention) are observed as outputs. Even though the retention candidate is not the best candidate, it is still accessible to language users via the rank-ordering. However, the accessibility of a candidate depends on the position it occupies in this rank-ordering – the higher position it occupies the more accessible it is. Since the deletion candidate occupies a higher slot than the retention candidate in both (8a) and (8b), the deletion candidate in both pre-vocalic and pre-consonantal context is more accessible than the retention candidate. The prediction is therefore that the deletion candidate will be observed as output more frequently than the retention candidate in both of these contexts.

Variation is possible because language users can also access non-optimal candidates. The relative frequency of different variants is accounted for by the fact that not all candidates are equally accessible. There is one complication here – in (8) I consider only two candidates. The generated comparison set for any input obviously contains many more than just two candidates. In §2.2.3 below I will come back to these other candidates. In §2.2.2 I first show how to account for the inter-contextual variation.

2.2.2 Inter-contextual variation

In the previous section I have shown how the rank-ordering model of EVAL can be used to account for the variation in the pronunciation of individual inputs. However, this is not the only relevant aspect of variation. We can now explain why deletion is preferred over retention in both pre-consonantal and pre-vocalic context. But deletion is preferred even more in pre-consonantal than pre-vocalic context – pre-consonantal context has a deletion rate of 85% while pre-vocalic context has a deletion rate of only 63%. We also need to account for this. In order to account for inter-contextual variation, I use the ability of EVAL to evaluate non-generated comparison sets.

The driving force behind the deletion of final [t, d] is to avoid violation of the markedness constraints *Ct#C and *Ct#V. Violation of these two constraints can be avoided by deletion, i.e. at the expense of a MAX-violation. The retention candidate in pre-consonantal position violates *Ct#C, while the retention candidate in pre-vocalic position violates *Ct#V. Because of the ranking ||*Ct#C o *Ct#V|| retention in pre-consonantal position is more marked than retention in pre-vocalic context. The drive to delete in pre-consonantal context is therefore stronger than in pre-vocalic context. This explains why pre-consonantal context has a higher deletion rate. This intuition can be captured formally by allowing EVAL to compare the faithful retention candidates from pre-vocalic and pre-consonantal contexts. This comparison is shown in (9).

(9) **Non-generated comparison set: the faithful candidates**

		*Ct#C	*Ct#V	MAX
2	/…Ct # t/ → [Ct # t]	*		
1	/…Ct # V/ → [Ct # V]		*	

Output of EVAL

Pre-vocalic: /…Ct # V/ → [Ct # V] $_{\text{*Ct\#V}}$

|

Pre-consonantal: /…Ct # C/ → [Ct # C] $_{\text{*Ct\#C}}$

The comparison in (9) shows that retention in pre-consonantal context is more marked than retention in pre-vocalic context. A greater decrease in markedness can therefore be bought by a MAX-violation (by deletion) in pre-consonantal than in pre-vocalic context. This explains why more deletion is observed in pre-consonantal than pre-vocalic context. By allowing EVAL to evaluate non-generated comparison sets, we can also explain why different contexts show different variation patterns.

2.2.3 Limiting variation: the critical cut-off

In the discussion of the intra-contextual variation §2.2.1 I considered only two candidates for each input. However, the generated comparison set for each input contains many more candidates than just these two. Under the rank-ordering model of EVAL, each of these candidates will occupy a slot in the rank-ordering. And under the assumption that the language user has access to levels below the highest level in this ordering, we predict that these other candidates should also be accessed as variants, even if less frequently. This problem becomes particularly acute when we consider candidates other than the deletion candidate that also avoid violation of the constraints *Ct#C and *Ct#V. In addition to deletion of the final [t, d], there are many other ways in which violation of these constraints can be avoided. For instance, it is possible to insert a vowel between the [t, d] and the preceding consonant. A phrase such as *west bank* can therefore be pronounced with deletion of the final [t] as [wɛs bæŋk], or with insertion of a vowel between [s] and [t] as [wɛsət bæŋk]. Both of these pronunciations avoid violating *Ct#C, both will be in the generated comparison set of /wɛst bæŋk/, and both will occupy a slot in the rank-ordering that EVAL imposes on this comparison set. Why is only the deletion candidate ever observed as a variant pronunciation for *west bank*?

Jamaican English tolerates deletion but not epenthesis in order to avoid violation of *Ct#C. Stated in terms constraint violation: Jamaican English is willing violate MAX but not DEP in order to avoid a violation of *Ct#C. Even though every candidate in the generated comparison set occupies a slot in the rank-ordering that EVAL imposes on this set, and even though the whole candidate set is in principle accessible to the language user via this rank-ordering, the language user will not access this rank-ordering to an arbitrary depth. There are certain constraints that a language is willing to violate – in the case at hand here those constraints are the markedness constraints *Ct#C and *Ct#V, and the faithfulness constraint MAX. But there are also constraints that a language is not willing to violate unless if absolutely required. From the discussion above, it follows that Jamaican English is not willing to violate DEP. I propose that there is a critical cut-off on the constraint hierarchy that divides the constraint set into those constraints that a language is willing to violate and those that a language is not willing to violate. A candidate disfavored by a constraint ranked higher than the cut-off will not be accessed as output if there is a candidate (or candidates) available that is not disfavored by any constraint ranked higher than the cut-off.

In Jamaican English both the deletion and the retention candidate are accessed as variant outputs. Both of these candidates are therefore disfavored only by constraints ranked lower than the critical cut-off. This means that MAX, *Ct#C and *Ct#V rank lower than the critical cut-off. The epenthetic candidate is never accessed as a variant output, and this candidate should be disfavored by at a constraint ranked higher than the cut-off. The epenthetic candidate violates the anti-epenthesis constraint DEP, and this constraint therefore ranks higher than the cut-off. In the tableau in (10) I reconsider the generated comparison sets from (8) above. However, this time I include the epenthetic candidate, the constraint DEP, and the critical cut-off. The critical cut-off is indicated by a thick vertical line in the tableaux. Constraints to the left of this line are ranked higher than the cut-off, and constraints to the right of this line are ranked lower than the cut-off. More discussion of the conventions used in these tableaux follows below the tableaux.

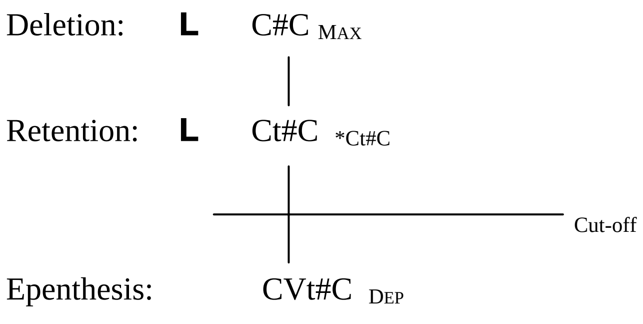
(10) **The generated comparison sets with epenthetic candidates**

Ranking: ||DEP o Cut-off o*Ct#C o*Ct#V o MAX||

a. **Pre-consonantal context**

/..Ct # C/	DEP	*Ct#C	*Ct#V	MAX
1 C#C				*
2 Ct#C		*		
3 CVt#C	*!			

Output of EVAL



Deletion: L C#C MAX

Retention: L Ct#C *Ct#C

Cut-off

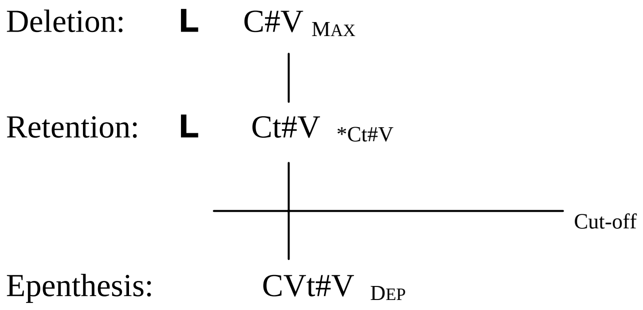
Epenthesis: CVt#C DEP

((10) continued)

b. **Pre-vocalic context**

/..Ct # V/	DEP	*Ct#C	*Ct#V	MAX
1 C#V				*
2 Ct#V			*	
3 CVt#V	*!			

Output of EVAL

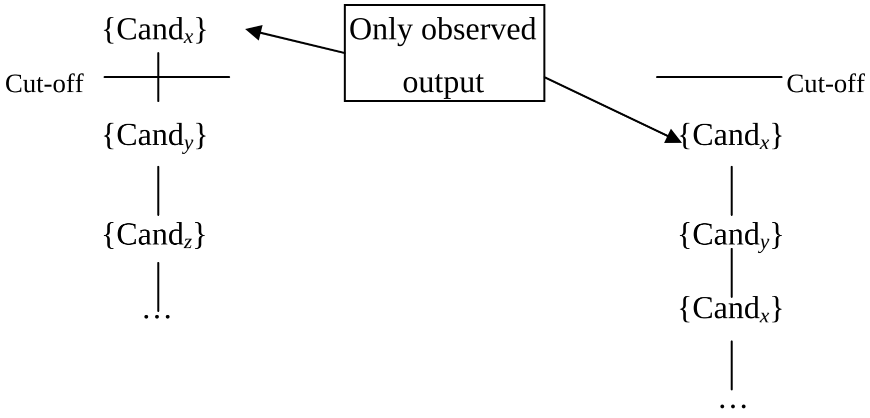


The epenthetic candidate violates DEP, which is ranked higher than the critical cut-off. Since there are candidates available that are not disfavored by any constrains ranked higher than the cut-off, the epenthetic candidate will not be accessed as a variant output. This is indicated in two ways in these tableaux. I use the exclamation mark to mark the DEP-violation of the epenthetic candidate. This exclamation mark has the same meaning as in classic OT tableaux – it indicates the violation that is responsible for eliminating a candidate as a potential output. On the graphic representation of the rank-ordering that EVAL imposes on the candidate set, I draw a solid horizontal line indicating the position of the critical cut-off. Candidates appearing higher than this line are not disfavored by any constraints ranked higher than the cut-off. Candidates appearing below this line are disfavored by at least one constraint ranked higher than the cut-off. Since there are candidates appearing above this line, the language user will not access any candidates below this line. The pointing hands marking the deletion and retention candidates indicate that they are both accessed as outputs. The epenthetic candidate does not have a pointing hand, indicating that it is not a possible output. The epenthetic candidate stands in here for all candidates other than the deletion and retention candidate – that is, all other candidates are also disfavored by at least one constraint ranked higher than the cut-off.

Although all candidates appear in the rank-ordering that EVAL imposes on the candidate set and although the language user has potential access to the complete rank-ordered candidate set, he/she will not normally access this rank-ordering to an arbitrary depth. In general, candidates disfavored by constraints ranked higher than the cut-off are not accessed.

Given this account of variation, how can categorical phenomena be accounted for? what conditions must be met for language users to access only the best candidate in the rank-ordering that EVAL imposes on the candidate set? The discussion in the rest of this section is general in nature and does not pertain specifically to the Jamaican data above. There are two ways in which a categorical phenomenon can arise: (i) If all but one candidate are disfavored by at least one constraint ranked higher than the cut-off, then the single candidate that is not disfavored by a constraint ranked higher than the cut-off will be the only observed output. This is exactly like a variable phenomenon, except that there is only one variant that is observed 100% of the time. (ii) It is also possible that all candidates are disfavored by at least one constraint ranked higher than the cut-off. When this happens the language user has no option but to access candidates that are disfavored by a constraint ranked higher than the cut-off. In order to minimize the number of such candidates that appear as outputs, only the single best candidate is accessed in such a situation. These two ways in which a categorical phenomenon can be modeled are represented graphically in (11).

(11) **Categorical phenomena in a rank-ordering model of EVAL**



2.2.4 Summary of proposal

I propose that variation should be explained in the following way: (i) EVAL imposes a harmonic ordering on the complete generated comparison set for every input, and the language user can potentially access all the candidates on this harmonic ordering. However, the accessibility of a candidate is directly related to the position it occupies in the rank ordering. The lower the position a candidate occupies in the rank-ordering, the less accessible it is. This explains why certain variants occur more often than others. The most frequent variant is the candidate rated best by the grammar, the second most frequent variant is the candidate rated second best, etc.

(ii) EVAL can compare morphologically unrelated forms (non-generated comparison sets). This property of EVAL is responsible for explaining why a variable process applies more or less frequently in different contexts. Suppose that the relevant non-generated comparison set contains the fully faithful candidates from two contexts, and that the fully faithful candidate of one context is more marked than the fully faithful candidate of another context. Then the drive to be unfaithful is stronger in the more marked context, and this context is then predicted to be subject to higher rates of application of the variable process.

(iii) There is a critical cut-off point on the constraint hierarchy of every language. The language user will not access a candidate that is disfavored by a constraint ranked higher than the critical cut-off point if a candidate is available that is not disfavored by any such constraint. This explains why variation is limited to a few variants per input, and also why there are some contexts in which no variation is observed.

3. Applying the rank-ordering model of EVAL to the phonological processing of non-words

In this dissertation I will argue that the rank-ordering model of EVAL can also be used to account for the role that grammar plays in the phonological processing of non-words. Specifically, it can explain (i) how grammar influences judgments on the well-formedness of non-words, and (ii) how grammar influences the reaction times in lexical decision tasks. There is a large body of literature on well-formedness judgments and lexical decision tasks (e.g. Balota and Chumbley, 1984, Berent and Shimron, 1997, Berent *et al*., 2001a, 2001b, 2002, Frisch *et al*., 2001, Frisch and Zawaydeh, 2001, Hayes, 1997, 1998, Hayes and MacEachern, 1996, 1997, Pierrehumbert *et al*., In press, Shulman and Davison, 1977, Stone and Van Orden, 1993, Vitevitch and Luce, 1999). Although much of this literature focuses more on the non-grammatical aspects of phonological processing, there is still a general acknowledgement that grammar does play some part in phonological processing. In particular, two generalizations that arise from this literature are that: (i) a non-word that is more well-formed according the grammar of some language is also judged to be more well-formed by speakers of that language, (ii) the less well-formed a non-word is according to grammar of some language, the more quickly speakers of that language are in detecting it as a non-word in a lexical decision task.

In this section I will illustrate briefly how the rank-ordering model of EVAL can be used to account for these generalizations. I will use data from well-formedness judgments and lexical decision experiments that I conducted with speakers of English. The purpose of the discussion here is not to give a full account of these data, but rather to illustrate how the rank-ordering model of EVAL can be used to account for this kind of data. A detailed discussion of the data can be found in Chapter 6 §3.

English allows words of the form [sTvT], but not of the form [sKvK] or [sPvP] – i.e. *state* is a word but **spape* and **skake* are not even possible words (Browne, 1981, Clements and Keyser, 1983, Davis, 1982, 1984, 1988a, 1988b, 1989, 1991, Fudge, 1969, Lamontagne, 1993: Chapter 6). I argue that there is a constraint against each of these types of forms – i.e. *sTvT, *sKvK and *sPvP. In English, *sKvK and *sPvP outranks some faithfulness constraint so that forms that would violate these constraints will never surface faithfully. However, *sTvT ranks lower than all relevant faithfulness constraints, so that forms that violate this constraint will surface faithfully. This therefore gives us the ranking ||{*sPvP, *sKvK} o Faithfulness o *sTvT||. On the grounds of cross-linguistic data and other phonotactic restrictions in English, I argue that there is also a ranking between *sPvP and *sKvK, namely ||*sPvP o *sKvK||. The complete ranking is therefore ||*sPvP o *sKvK o Faithfulness o *sTvT||.

I conducted an experiment in which I presented subjects with pairs of non-words of the form (i) [sTvT]~[sKvK], (ii) [sTvT]~[sPvP], and (iii) [sKvK]~[sPvP]. The task of the subjects was to select from each pair the non-word that is most well-formed. In the [sTvT]~[sKvK]-pairs subjects chose [sTvT] more often (75% of time), in the [sTvT]~[sPvP]-pairs they also chose [sTvT] more (78% of the time), and in the [sKvK]~[sPvP]-pairs they selected [sKvK] more (55% of the time). In general, the response pattern can therefore be summarized as: [sTvT] > [sKvK] > [sPvP]. This corresponds exactly to the assumed ranking between the three *sCvC constraints – non-words that violate the lowest ranked *sTvT constraint are chosen most frequently, then non-words that violate the second highest ranked constraint *sKvK, while non-words violating the highest ranked *sPvP are chosen least frequently.

I also conducted a lexical decision experiment. In this experiment I presented English speakers with a list of words and non-words, and their task was to decide for each token whether it is a word or a non-word. The non-words included [sTvT]-, [sKvK]- and [sPvP]-forms. I recorded the reaction times on correct non-word responses for these three kinds of non-words. The finding was that [sTvT] non-words were detected most slowly (mean response time = 403.53 ms), [sKvK] non-words more quickly (mean response time = 350.45 ms), and [sPvP] non-words the most quickly (mean response time = 303.33 ms). This also corresponds to the ranking between the *sCvC-constraints. The higher ranked the markedness constraint that a non-word violates, the more quickly it is identified as a non-word.

These response patterns can be explained by allowing EVAL to compare non-words of the form [sTvT], [sKvK] and [sPvP]. Such a comparison is shown in (12).

(12) **Comparing [sTvT], [sKvK] and [sPvP]**

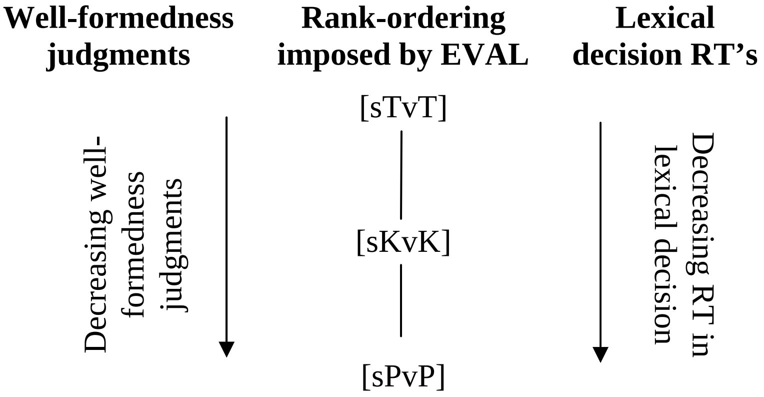
	*sPvP	*sKvK	Faithfulness	*sTvT
sTvT				*
sKvK		*		
sPvP	*			

Output of EVAL

[sTvT] *sTvT
|
[sKvK] *sKvK
|
[sPvP] *sPvP

The higher a form appears in the rank-ordering that EVAL imposes on the comparison set, the more well-formed it is judged to be by language users. Similarly, the higher a form appears on this rank-ordering, the more seriously it is considered as a possible word, and the longer it takes to identify it as a non-word. The relation between this rank-ordering, and well-formedness judgments and lexical decision reaction times is represented graphically in (13).

(13) **Rank-ordering model of EVAL, well-formedness judgments and lexical decision reaction times**


Well-formedness judgments
Rank-ordering imposed by EVAL
Lexical decision RT's
[sTvT]
[sKvK]
[sPvP]
Decreasing well-formedness judgments
Decreasing RT in lexical decision

Recall that I am proposing two changes in the way that we think about EVAL: (i) First, I assume that EVAL imposes a harmonic rank-ordering on the full comparison set, and does not only distinguish the best candidate from the rest. (ii) Secondly, I assume that EVAL can evaluate candidates that are not morphologically related (non-generated comparison sets). Both of these assumptions are used in explaining the [sCvC]-data.

First, the three forms that are being compared in the tableau in (12) are not morphologically related. EVAL is therefore evaluating a non-generated comparison set here. Had EVAL only been able to compare candidate sets generated by GEN for some input, this comparison would not have been possible at all. Secondly, the output of EVAL in (12) consists of a three-level ordering. EVAL does more than distinguish between the best [sTvT] candidate and the other two candidates. EVAL also orders the two non-best candidates relative to each other. Since the subjects in the experiment treated [sKvK]- and [sPvP]-forms differently, we need a grammar that can distinguish between them. EVAL therefore generates information even about how the non-best (losing) candidates are related to each other, and this information is available to and accessed by language users.

4. Structure of the dissertation

This dissertation contains five content chapters. Chapter 2 is theory oriented. In this chapter I develop a set theoretic model of EVAL. This chapter is somewhat independent from the rest of the dissertation. It contains many general results about an OT grammar of which only some are directly relevant to this dissertation. The results that are relevant are: (i) I show that the rank-ordering model of EVAL is completely compatible with standard assumptions about an OT grammar. The rank-ordering model of EVAL does not require any changes or additions to the architecture of an OT grammar. (ii) I also show that nothing in the way that EVAL works depends on the origin of the set of candidates that is being compared. Allowing EVAL to compare non-generated candidate sets therefore does not require any additions to the architecture of an OT grammar. This chapter will not be easily accessible to the non-mathematically inclined reader. I would suggest that these readers skip Chapter 2 and start reading at Chapter 3.

The rest of the dissertation illustrates empirical applications of the rank-ordering model of EVAL. In Chapters 3 to 5 I illustrate the application of this model to phonological variation. I discuss three examples: (i) Chapter 3 deals with variable vowel deletion in Latvian, (ii) Chapter 4 deals with variable vowel deletion in Faialense Portuguese, and (ii) Chapter 5 deals variable deletion of word-final coronal stops in English. Chapter 3 is a short chapter and serves an introductory purpose. In this chapter I explain the theoretical assumptions that I make in more detail. I suggest that readers start with this chapter. However, Chapters 4 and 5 are independent from each other and can be read separately.

In Chapter 6 I apply the rank-ordering model of EVAL to the phonological processing of non-words. I discuss two examples: (i) How the OCP influences well-formedness judgments and lexical decision in Hebrew, and (ii) how *sCvC-constraints influence well-formedness judgments and lexical decision in English.