Combining Optimality Theory and a Cognitive Architecture

Jan M.V. Misker (J.M.V.Misker@its.tudelft.nl)
Delft University of Technology, Department of Information Technology and Systems,
Mekelweg 4, 2628 CD Delft, The Netherlands

John R. Anderson (ja+@cmu.edu)
Carnegie Mellon University, Department of Psychology, 5000 Forbes Avenue
Pittsburgh, PA 15232, USA

Abstract

Optimality Theory (OT) has had a lot of attention from the linguistics research community but also still largely lacks cognitive grounding. We used the ACT-R cognitive architecture to gain greater insight into the cognitive grounding issues that OT needs to address, most notably the GEN process and the learning of the constraint ranking. A generic ACT-R 5.0 model was developed guided by OT principles. The generic model was instantiated in two specific models, one for syllabification and one for past tense formation. Realistic perception data was used to train the models, both were successful in learning the correct constraint ranking for their domain. This result partly bridges the gap between Optimality Theory and ACT-R, providing OT with a better cognitive grounding and ACT-R with better linguistic capabilities.

Introduction

Overview of ACT-R

ACT-R (Anderson & Lebiere, 1998) is a cognitive architecture. As such it provides a framework for developing computational models of a wide variety of cognitive tasks. These models are constrained by the architecture in the way they retrieve, store and process information. Declarative knowledge is stored in memory as chunks that can contain only a limited amount of information, either as atomic values or as references to other chunks. Procedural knowledge is represented by productions; IF-THEN rules that compete with each other. Every production cycle (50ms), one rule is selected to alter the current goal, which is a chunk.

Overview of Optimality Theory

In the early 1990's, Optimality Theory was introduced to the linguistics research community (Prince & Smolensky, 1993). The central idea in OT is the concept of *Harmony*; a measurement to determine how well an utterance fits with the grammar of a language, expressed in a set of constraints. Figure 1 shows the entire process that OT proposes to take place when producing an utterance.

The *input* is the lexical representation of a word, the *output* is the final utterance as spoken by the speaker. When an input is selected to be uttered, a function called *GEN* will generate a set of potential utterances, called the candidate set. The *EVAL* function will then evaluate which of these candidates is the most optimal one, given an ordered set of constraints (*CON*). A constraint specifies a certain condition, and a constraint violation means that that condition is not satisfied in the candidate. For example, the constraint called FaithV is a faithfulness constraint. It specifies that each vowel in the input should be present in the output as well. Every omission or addition is a violation of the constraint

The EVAL function can best be described by splitting it up in two steps: (1) first for each candidate the highest-ranking constraints that is violated by that candidate is determined. The result of this step is that each candidate is now accompanied by a constraint violation. (2) Next, these constraint violations are compared to determine which candidate violates (if any) the constraint that is ranked lowest, this candidate is the most optimal candidate and therefore the output.

An important aspect of OT is that the constraint set is assumed to be universal, i.e. the same for all languages. Only the ranking of the constraints is different for each language.
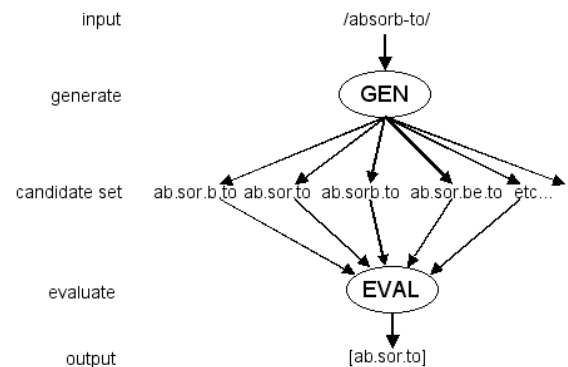

Figure 1: The Optimality Theory process.

Table 1: Example tableau showing Spanish syllabification.

Spanish: FaithV, Peak, *Complex >> FaithC

/absorb-to/	FaithV	Peak	*Complex	FaithC
☞ ab.sor.to				*
ab.sorb.to			*!	
ab.sor.be.to	*!			
ab.sor.b.to		*!		

Table 2: Example tableau showing English syllabification.

English: FaithV, FaithC, Peak >> *Complex

/soft-nɛs/	FaithV	Peak	FaithC	*Complex
☞ soft.nɛs				*
sof.nɛs			*!	
sof.ti.nɛs	*!			
sof.t.nɛs		*!		

Example. Table 1 and Table 2 (adopted from Archangeli, 1997), provide an example of how OT explains certain characteristics of the syllabification of words. The vowel in a syllable is called the nucleus, the consonants before it are called the onset and the consonants after it are called the coda.

The constraints used in this example are FaithV, FaithC, Peak and *Complex. FaithV and FaithC are faithfulness constraints for vowels and consonants, respectively. Peak is violated when a syllable does not contain exactly one vowel and *Complex is violated when the onset or coda is *complex*, i.e. when they consist of more than one consonant each.

The first column contains the input, in the top row, and (part of) the candidate set, the next columns each correspond to one constraint. The ranking of the constraint is reflected in the order of the columns. A dotted line between two columns indicates that the ranking of those two constraints is equal or not important to the phenomenon currently being explained. Constraint violations are marked with asterisks (*). An exclamation mark (!) indicates a *fatal* violation. The row that corresponds to the optimal candidate is marked with a hand symbol.

Table 1 shows that in Spanish sometimes a consonant disappears from the output, in this case in an adjective, indicated by the "-to" suffix. The OT explanation for this is that in Spanish the *Complex constraint is ranked higher than the FaithC constraint. Table 2 shows that in English consonants do not disappear, because FaithC is ranked higher than *Complex.

Combining OT & ACT-R

Why combine?

The interest in combining the two theories is twofold. First of all, it is expected that by applying insights into cognition, as used in the ACT-R framework, the cognitive basis for Optimality Theory can be enhanced. Second, ACT-R will be a better cognitive architecture if it can also explain linguistic phenomena.

Improving the cognitive basis of OT. OT has a number of open issues regarding its cognitive basis. Although OT is connected to connectionism, through its roots in Harmony Theory, this connection does not satisfactory explain how OT might be implemented in the brain. Harmony Theory does support the EVAL process, but both the GEN process and the cognitive/neural basis of the candidate set are underspecified. These two issues are of course deeply connected, because GEN generates the candidate set.

The GEN process needs to generate a candidate set that in some cases has to be very irregular. For example the candidate set that contains the past tense of the English irregular verb "go" needs to contain the word "went". Furthermore, the candidate set is sometimes required to be very large, but how the brain would create and analyze such a vast set is left unexplained.

ACT-R needs to speak up. The ACT-R theory still lacks a usable speech/linguistic module. In order to explain certain phenomena, for example language acquisition, detailed knowledge about the processes involved in perception and production of speech is required.

Design

The high level design of the ACT-R model of OT is as follows: (a) The input is represented by strings of characters, which are a scaled down version of the IPA phonetic alphabet. (b) Constraints are represented by production rules, competing with each other based on their expected gain. This competition represents the ranking of the constraint set. (c) The GEN function is modeled by finding analogies between the current input and previous inputs. The transformation needed to get the output out of the analog input is then applied to the current input. (d) The EVAL function is modeled by comparing the candidate generated by GEN with the most optimal form until that time in the process. (e) The EVAL function iterates over the candidates generated by GEN, either until no new candidates are formed or until a certain fixed number of iterations have been processed. (f) The model can operate in two modes: perception or production. In perception mode the perceived form is considered the optimal form and can be used to learn the order of the constraints. In production mode the form that the model considers best is generated.

The candidate set is thus never represented entirely, only the best candidate so far is stored. A competing candidate is generated and compared with the current best.

Learning

Learning in OT. Two aspects of OT involve learning, the constraint ranking and the GEN function/lexicon. Learning in the GEN function and/or the lexicon has not received the same attention in the OT literature as the learning of the constraint ranking has. Tesar and Smolensky (2000) argue that learning the constraint ranking is accomplished as follows: given a perceived utterance, it is assumed that the input from which that utterance was derived is known This input is used to generate a candidate set, which is evaluated against the current constraint ranking. If a candidate is more optimal than the perceived utterance, this indicates that the current constraint ranking is incorrect, given that the perceived form is the most optimal one. In such a case, the ranking is adjusted by demoting the constraints that are violated by the perceived form but not by the generated form until the candidate is not evaluated as more optimal than the perceived utterance anymore. Other approaches (Boersma 2000; Stemberger and Bernhardt 2001) also allow for promoting constraints.

Learning in OT + ACT-R. There are several possibilities for learning in a combined OT and ACT-R framework. The focus in this research has been on applying learning mechanism inherent in ACT-R to the learning issues of OT. By identifying constraints with

productions, the ACT-R learning mechanisms for the expected gain of a production make it possible to learn the constraint ranking by its mechanisms for learning production ranking. ACT-R also provides learning mechanisms that might be used in learning involved in the GEN process but these learning mechanisms are not considered in the current design.

Detailed description of the model

The chunk type. Chunks form the declarative knowledge in ACT-R. Only one type of declarative knowledge is used throughout the model, it serves as the goal, and old goals are stored in memory. These goal chunks contain the input, the optimal form, a competing candidate, and some bookkeeping information. All perceived and produced utterances are stored as chunks, since they were goals once.

Constraint implementation. Every constraint is associated with one production rule. The constraints are implemented as external functions in Lisp code, called by the production. The reason for this is that the operations require rather more elaborate calculations than could only be carried out by ACT-R in a single production cycle. We think of these as representing special capabilities of a speech module.

Flow of control. The different states of the entire process are illustrated in Figure 2. We will discuss each of the three components and the iteration illustrated there:

GEN. The GEN process generates analogies in two steps. (1) In the first step a request is sent to ACT-R's retrieval process to retrieve a chunk with an input value similar to the input value of the goal. In the next step of the analogy process (2) the retrieval buffer is examined. If it is empty the entire process is terminated because there are no more similar candidates. If it contains a memory chunk, this chunk is analyzed to determine the transformation that alters its input slot to its output slot. The transformation is then applied to the input of the current goal and the result is stored in the candidate slot of the goal.

EVAL. The candidate and current best options are stored in the goal. Different productions representing different constraints propose either the candidate or the optimal form as best form, according to the constraint violations in both. ACT-R will select the production with the highest expected gain to place the best form in the best slot of the goal. The expected gain can be learned by ACT-R from experience with that production, described below. The production that fires thus represents the highest-ranking constraint that is violated by either the optimal or candidate form.
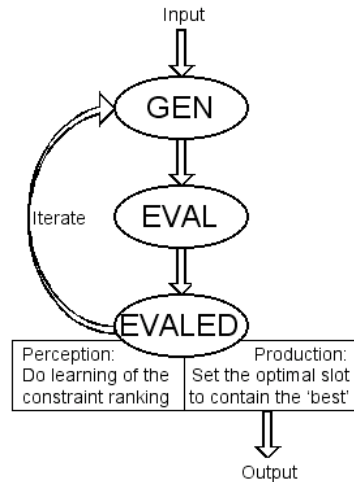
Figure 2: Flow of the ACT-R model of OT.

EVALED. Only when the state has reached EVALED, the model behaves different depending on the mode of operation; perception or production. This is an important aspect of the model, because it is not necessary to devise two versions of GEN, and the learning of the constraint ordering that occurs in perception mode is directly reflected in production mode.

In perception mode the best slot is compared with the optimal and candidate slot. If the candidate is the best, this means that the ordering of the constraints is wrong, i.e. the constraint that indicated that the competing candidate form is more optimal than the perceived optimal form should be ranked lower. The production that fires on this situation is marked with the failure attribute. If it fires this results into an increase of the number of failures for all preceding productions, thus including the production associated with the constraint that is ranked too high. A higher number of failures leads to a decrease of the expected gain value (PG-C) of that production which makes it less likely that that production will fire again in that context in the future. If the optimal (perceived) form is selected as the best, the success attribute is marked, leading to an increase of the expected gain value. The end result of this learning process will be a stable ranking of the constraints, ideally reflecting the correct ranking.

When the goal is to produce an utterance, the optimal slot simply is replaced with the value of the best slot. The expected gain values are thus not adjusted in this case.

Iterations. The above three components could generate and evaluate the entire candidate set. Two issues prevent such behavior: (1) It is unknown how large the candidate set is, in fact in this model the generate function may continue to generate indefinitely. (2)

After a certain amount of time, something will have to be uttered. For these reasons, only a fixed, small, number of iterations are allowed in the model.

Two applications: syllabification and past tense

Two linguistic phenomena have been modeled in detail, syllabification and past tense formation, both only in English. Syllabification is the process by which sequences of phonemes are divided into syllables. The formation of the past tense of verbs is a widely researched phenomenon, especially regarding the U-shaped learning that is involved. The six syllabification constraints are slightly modified versions from Archangeli (1997), their meaning is explained in the example above. The five past tense constraints are similar to those in Stemberger and Bernhardt (2001).

Data

Syllabification data. A data set comparable to what children are exposed to is used to train the model. The data is gathered from the CHILDES database according to the following criteria: (a) Only utterances made by the mother or father of the child are used. (b) The child addressed is not older than 2 years and 6 months. (c) The utterances are part of a free speech session, as opposed to a laboratory dialogue. The data set is assumed to give a reasonably fair representation of the utterances a child is exposed to. The following data sets met these criteria, sometimes partially to stay within the age requirement, and were used to form the general data set: Bernstein (1982), Demetras (1989a, 1989b), Higginson (1985), Howe (1981) and Korman.

Of the resulting data set the thousand most frequent words were transcribed in the rough phonetic transcription used in the model, along with the syllabic structure of the word. It is noteworthy that in this sample 87.5% of the word forms uttered have one syllable, 11.5% have two syllables and only 1% has more than two syllables. Given the high ranking of Faithfulness and Peak in English, the *Complex, NoCoda and Onset constraints are only of interest for word forms with multiple syllables, which are not very frequent. The data sample is presented to the model one by one as perceived speech utterances, according to the frequency distribution.

Past Tense data. The data used to train the past tense model is the same as the data used in Taatgen and Anderson (2002). This data set contains 478 verbs that children or their parents use as reported by Markus et al. (1992) with frequency information from Francis and Kucera (1982). The first person present and past tense of these verbs were transcribed in the rough phonetic form used also in the syllabification model and

presented to the model as perceived utterances, distributed according to their real world frequency.

Noteworthy is that the 13 most frequent verbs, accounting for 65% of the total based on frequency, are irregular verbs[1], with "am"/"was" alone accounting for 35%. This will lead to problems in the learning of the constraint ranking because the constraint set cannot qualify these irregular forms as the most optimal ones in all cases.

Contrary to syllabification, there is data available on how children perform on the forming of past tenses. This data applies to the errors children make in their use of a regularized past tense when the actual past tense is irregular. Such data shows that children exhibit U-shaped learning, separable in three stages. In the first stage irregular and regular forms are used correctly. In the second stage the irregular forms are regularized; i.e. forms like "*kEped" or "*havd" might be uttered. In the third stage the correct forms are used again for both regular and irregular past tenses.

Results

Syllabification results. The graph in Figure 3 shows the expected gain values of the productions associated with the constraints after every time the model has perceived and utterance. It clearly shows that the constraint ranking converges to the correct ranking rather quickly. This graph depicts a typical run of the model with 1000 exposures to syllabification data. After such a run the production performance of the model is about 95%.

The most promising result is that the correct constraint ranking is in fact learned.

Past Tense results. As with the syllabification model, the correct constraint ranking is learned, shown in Figure 4. The production performance of the model is not very good, after a typical run of 1000 perceptions the production performance is about 2/3 correct.

U-shaped learning is not modeled, because the current model uses a constraint set that cannot deal with exceptions. For example if both "amd" and "was" are generated as candidates for "am", "amd" is the optimal form according to this constraint set and ordering. The only way to resolve this issue is by bypassing the constraint evaluation, which might be achieved by proceduralization, as discussed in the final discussion and conclusion section.
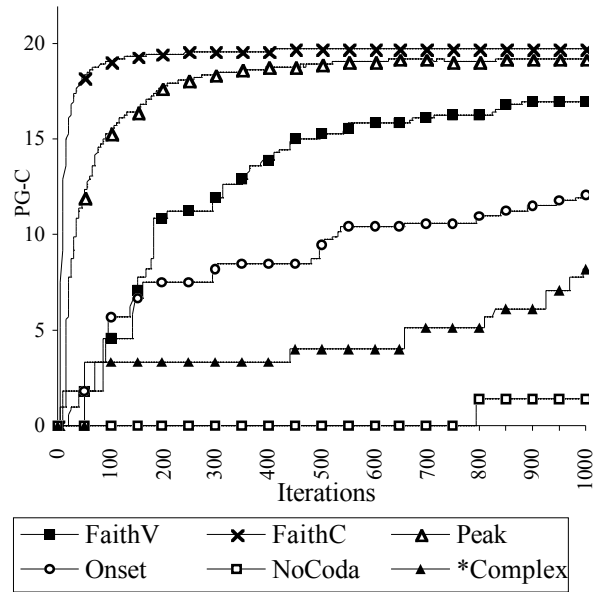

Figure 3: The ranking of the constraints in the syllabification example, as identified by the expected gain values of the ACT-R productions.
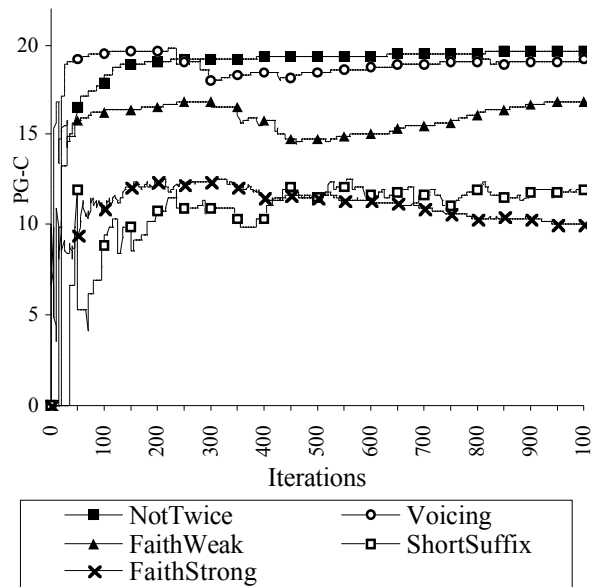

Figure 4: The ranking of the constraints in the past tense example.

Discussion and conclusions

Both the models are able to learn the correct constraint ranking, which is the most important and promising result. The models however do not yet show realistic human/child like speech behavior. This limits the ability to validate the results on the empirical level. The models are best thought of as models of early language development. Adult performance might be achieved through proceduralization; an ACT-R mechanism that

---

[1] On closer inspection, irregularity is not such a well-defined concept. Burzio (2002) argues that from a phonetic viewpoint the supposedly irregular past tense of for example "keep", being "kept", can be regarded as more phonetically regular than the regular form "*keeped".

gradually moves knowledge from declarative memory (chunks) to procedural memory (productions). Proceduralization can even bypass the generation/evaluation process entirely by introducing productions specific for uttering a certain morpheme, word or even parts of sentences. This will allow for exceptional cases to be handled correctly as well. Note that learning of the constraint ranking occurs only from perceived utterances and so it can start well before a child starts producing utterances herself.

We also do not treat in detail the nature of the individual constraints. The current model treats each constraint as external to central cognition, linked in through its associated productions. OT only claims that the constraints are universal, not what their nature is. More research is needed to determine whether it is possible to learn the constraints themselves also, besides just their ranking. From an ACT-R viewpoint this seems possible. Another possibility is that the constraints are the cognitive counterpart of physical conditions, for example the muscles in the vocal tract. This would account for the universal manifestation of these constraints. However that would have to be proven for every constraint separately. It could be acceptable from an ACT-R viewpoint, because ACT-R has to take into account the various limitations imposed on cognition by being embodied.

On a more theoretical level, the approach as described in this paper forms a bridge between two previously unrelated theories. To do this, Optimality Theory had to be adjusted to fit into the ACT-R architecture: the generate function, the candidate set and the constraint ranking learning mechanism were altered, all based on the limitations imposed by ACT-R. Negatively put, this can mean that one of the theories was wrong, since it was incompatible with a theory that dealt with the same phenomena. Of course a more constructive approach is to say that both theories need to converge to a more encompassing unified theory of cognition.

Acknowledgments

References

Anderson, J. R. & Lebiere, C. (1998). *The Atomic Components of Thought*. Mahwah, NJ: Lawrence Erlbaum Associates.

Archangeli, D. (1997). Optimality Theory: An Introduction to Linguistics in the 1990s. In Archangeli, D. & Langendoen, D. T. (Eds.), *Optimality Theory: An Overview*. Oxford, UK: Blackwell Publishers.

Bernstein, N. (1982). *Acoustic study of mothers' speech to language-learning children: An analysis of vowel articulatory characteristics* Unpublished doctoral dissertation. Boston University.

Boersma, P. (2000). Learning a Grammar in Functional Phonology. In Dekkers, J., Leeuw, F. van der, & Weijer, J. van de (eds.), *Optimality Theory: Phonology, Syntax, and Acquisition*. Oxford, UK: Oxford University Press.

Burzio, L. (2002). Missing players: Phonology and the Past-tense Debate. *Lingua* 112.

Demetras, M. (1989a). *Changes in parents' conversational responses: A function of grammatical development*. Paper presented at ASHA, St. Louis, MO.

Demetras, M. (1989b). *Working parents' conversational responses to their two-year-old sons*. Working paper. University of Arizona.

Francis, W.N., & Kucera, H. (1982). *Frequency analysis of English usage: lexicon and grammar*. Boston, MA: Houghton Mifflin.

Higginson, R.P. (1985). *Fixing-assimilation in language acquisition*. Unpublished doctoral dissertation. Washington State University.

Howe, C. (1981). *Acquiring language in a conversational context*. New York, NY: Academic Press.

Marcus, G.F., Pinker, S., Ullman, M., Hollander, M., Rosen, T.J., & Xu, F. (1992). Overregularization in language acquisition. *Monographs of the society for research in child development, 57*(4), 1-182.

Prince, A. & Smolensky, P. (1993). Optimality Theory: Constraint interaction in generative grammar. Technical Report CU-CS-696-93, Department of Computer Science, University of Colorado at Boulder, and Technical Report TR-2, Rutgers Center for Cognitive Science, Rutgers University, New Brunswick, NJ. April.

Stemberger, J.P., & Bernhardt, B.H. (2001). U-shaped learning in language acquisition, and restrictions on error correction. Poster presented at the 2001 Biennial Meeting of the Society for Research in Child development, April 22, 2001. ROA-472.

Taatgen, N.A., & Anderson, J.R. (2002). Why do children learn to say "broke"? A model of learning the past tense without feedback. *Cognition* 86(2).

Tesar, B, & Smolensky, P. (2000). *Learnability in Optimality Theory*. Cambridge, MA: MIT Press.